DOCUMENT RESUME

ED 055 252 VT 014 134

AUTHOR Magisos, Joel H.; Berkebile, Lonnie A.

TITLE Computer Applications to ERIC Clearinghouse Document Collections: A Final Report.

INSTITUTION Ohio State Univ., Columbus. Center for Vocational and Technical Education.

PUB DATE Mar 71

NOTE 323p.

EDRS PRICE MF-$0.65 HC-$13.16

DESCRIPTORS *Computer Programs; Data Bases; Information Centers; Information Networks; *Information Processing; *Information Retrieval; Information Systems; *Information Utilization; Manuals; *Program Development; Research Tools; Research Utilization; Vocational Education

IDENTIFIERS ERIC Clearinghouse Document Collections

ABSTRACT

The purpose of this project was to develop computer techniques for the production of VT-ERIC publications and to report these techniques so that they might be used by other ERIC Clearinghouses for production of their publications. The Clearinghouse needed to: (1) generate bibliographies from its own collection upon which to base information analysis papers, (2) conduct searches of the collection for solution to problems in the field, (3) generate camera-ready copy for publication of Clearinghouse abstract journals, and (4) describe the system to facilitate its installation in Clearinghouses and other agencies. Included in the report are an explanation of the system requirements, a description of computer program operating procedures, capability, and utilization, and a summary and recommendations. Extensive appendixes provide the technical details for the system including keypunching, job control language, control cards, error messages, program listing, program and system flow charts, and page samples. A related document, Computer Applications to ERIC Clearinghouse Document Collections: A User and Installation Handbook is available as VT 014 135. (Author/JS)

ED055252

VT014134

ERIC CLEARINGHOUSE
THE CENTER FOR VOCATIONAL
AND TECHNICAL EDUCATION
THE OHIO STATE UNIVERSITY
COLUMBUS, OHIO

U.S. DEPARTMENT OF HEALTH,
EDUCATION & WELFARE
OFFICE OF EDUCATION
THIS DOCUMENT HAS BEEN REPRODUCED EXACTLY AS RECEIVED FROM THE PERSON OR ORGANIZATION ORIGINATING IT. POINTS OF VIEW OR OPINIONS STATED DO NOT NECESSARILY REPRESENT OFFICIAL OFFICE OF EDUCATION POSITION OR POLICY

ED055252

COMPUTER APPLICATIONS TO ERIC CLEARINGHOUSE DOCUMENT COLLECTIONS: A FINAL REPORT.

by

Joel H. Magisos and Lonnie A. Berkebile

ERIC Clearinghouse on Vocational and Technical Education
The Center for Vocational and Technical Education
The Ohio State University
Columbus, Ohio 43210

March 1971

PREFACE

The ERIC Clearinghouse on Vocational and Technical Education (VT-ERIC) has been producing two quarterly abstract journals since late 1967. Abstracts of Research and Related Materials in Vocational and Technical Education (ARM) provides indexes to and abstracts of documents in vocational and technical education; it is designed for researchers, teacher educators, administrators, and others who need research and related materials in an organized and accessible form. Abstracts of Instructional Materials in Vocational and Technical Education (AIM) provides indexes to and abstracts of instructional materials in vocational and technical education; it is designed to provide accessibility to the collection for teachers, administrators, curriculum specialists, and others. Nearly 36 percent of the materials in ARM and 79 percent of those in AIM have not appeared in Research in Education and are made available only in this way to over 2,800 users who receive the publications.

The application of computer technology to the generation of AIM and ARM has afforded the Clearinghouse with economies and efficiencies otherwise not possible and has opened up new opportunities to utilize the Clearinghouse collection such as document retrieval and information product repackaging. Other Clearinghouses may find these techniques applicable to their operation. Agencies, such as state vocational-technical education research coordination units, will soon be able to acquire the ARM and AIM collection on magnetic tape and be able to retrieve document resumes with QUERY. VT-ERIC will continue to use and update the files.

Recognition should be given to Joel H. Magisos, coordinator of information services at The Center, and Lonnie A. Berkebile, computer specialist at The Center, for their work in planning, conducting, and reporting the project.

Robert E. Taylor
Director
ERIC Clearinghouse on Vocational
and Technical Education

INTRODUCTION

The project reported herein was conducted by the ERIC Clearinghouse on Vocational and Technical Education pursuant to an extension of its contract. The purpose of the project was to develop computer techniques for the production of Abstracts of Research and Related Materials in Vocational and Technical Education (ARM) and Abstracts of Instructional Materials in Vocational and Technical Education (AIM) and to report these techniques so that they might be used by other ERIC Clearinghouses for production of Clearinghouse publications. The specific objectives of the project were to enable the Clearinghouse to:

1. generate bibliographies from the Clearinghouse collection upon which to base information analysis papers,
2. conduct searches of the Clearinghouse collection for solution to problems in the field,
3. generate camera-ready copy for publication of Clearinghouse abstract journals (i.e., ARM, AIM, and AIM-ARM Annual Index), and
4. describe the system to facilitate its installation in Clearinghouses and other agencies.

Figure 1 on page vi illustrates a general outline of the system. A schematic flowchart of the programs is provided in Figure 2, page 18 - 21.

The document collection of the ERIC Clearinghouse on Vocational and Technical Education has been made available through announcement in USOE's monthly Research in Education (RIE) and through its own quarterly abstract series, Abstracts of Research and Related Materials in Vocational and Technical Education (ARM) and Abstracts of Instructional Materials in Vocational and Technical Education (AIM).

Efficiency and accuracy of ARM and AIM production has been gained through the application of computer techniques. Bibliographic data and abstracts are brought together on data cards for input to magnetic tape files. The magnetic tape file provides storage of the data. The computer programs developed in this project permit accurate manipulation of the data into formats for resume and index sections of AIM and ARM which may be printed out as camera-ready copy. Production of the Fall 1970 issue of ARM represented the first Clearinghouse publication which had been completely computer-generated. In addition, the Fall 1970 ARM contained the Clearinghouse's backlog and was released on schedule.

In the first twelve issues of ARM, 3,703 documents were announced, 1,324 of which had not been reported in RIE. In the first twelve issues of AIM, 2,605 documents were announced, of which 2,067 had not been reported in RIE. Thus, 3,391 of the documents announced in AIM and ARM could not be searched by computer until after the development of the software package reported here. The first search of a partial file has been successfully conducted and input of resumes in back issues of AIM and ARM to magnetic tape is nearly complete. A significant by-product of the project will be the eventual availability of the AIM and ARM collection on magnetic tape in a format to permit others to search it with QUERY.

The report includes explanation of the system requirements; descriptions of computer program operating procedures, capability, and utilization; and a summary and recommendations. Extensive appendices provide the technical details for the system including keypunching, job control language, control cards, error messages, program listings, program and system flowcharts, and page samples. Also available is Computer Applications to ERIC Clearinghouse Document Collections: A User and Installation Handbook by Lonnie A. Berkebile.

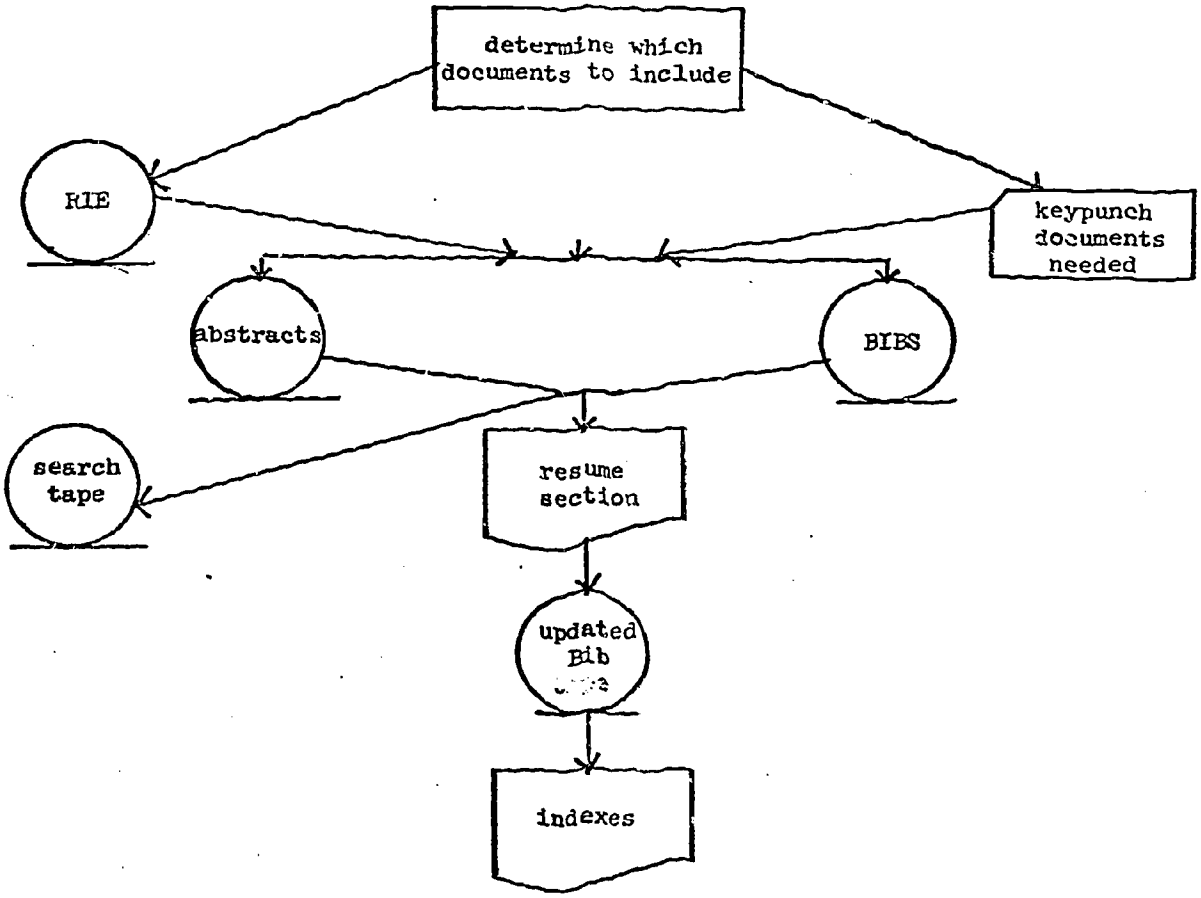


Figure 1 -- General Outline of the System

TABLE OF CONTENTS

PREFACE ii

INTRODUCTION iv

TABLE OF CONTENTS vii

LIST OF FIGURES viii

Chapter

I. SYSTEM REQUIREMENTS FOR EFFECTIVE COMPUTER UTILIZATION 1

Equipment Requirements for Clearinghouse Computer Programs
Storage of Data Pertaining to Clearinghouse Documents
Bibliographic Record Format
Production of Clearinghouse Information Products
Retrieval of Clearinghouse Documents

II. DESCRIPTION OF COMPUTER PROGRAM OPERATING PROCEDURES, CAPABILITY, AND UTILIZATION IN ERIC CLEARINGHOUSES 8

Indexes
The Document Number Index
Supporting Services Index
Subject Index
Author Index
Conversion of Document Number Index
Edit Programs
Master Edit Programs
Annual Subset Edit Programs
Use of QUERY Program

III. SUMMARY AND RECOMMENDATIONS 16

Summary of the Project
Recommendations

APPENDICES 1

LIST OF FIGURES

Figure		page
1.	General Outline of the System	vi
2.	Schematic Flow Chart of Computer Programs	18

SYSTEM REQUIREMENTS FOR EFFECTIVE COMPUTER UTILIZATION

The computer software that was developed in this project permits effective utilization of the computer in the production of Clearinghouse abstract journals such Abstracts of Research and Related Materials in Vocational and Technical Education (ARM) and Abstracts of Instructional Materials in Vocational and Technical Education (AIM). It also permits computer search of the document collection which is announced in the abstract journal. The principal benefits are increase in production efficiency and accuracy, access to the Clearinghouse document collection by computer, and increase in the options available for repackaging the document collection in other useful forms.

The following sections briefly describe the underlying requirements for, and general nature of, the procedures for data storage, publication production, and search tape preparation.

The first task was to prepare a package which could easily be converted to other computing operations and still conserve time in the program operation. Possible languages to be used were FORTRAN, ASSEMBLER, or PL/I. The first two languages have requirements which should be avoided. FORTRAN has limited character manipulation and presents difficulties with variable length records. ASSEMBLER would be difficult to modify and update. Thus, a decision was made to write the programs in PL/I, even though the package would be restricted to IBM installations. It might be noted, however, that other manufacturers are now producing PL/I compilers.

Equipment Requirements for Clearinghouse Computer Programs

The programs in this package have been run on an IBM 360 model 75 operating nder HASP-II. All of the programs in this package will fit in 126,000 bytes of ain memory. The running times that we have experienced indicate that 2 hours f computer time for each publication will be needed. The 2 hour estimate is for publication of 300 documents.

The equipment needed for producing an abstract journal and magnetic search ape is available at most computer centers with an IBM 360 computer. If both upper and lower case letters are used, each document will require approximately 0 minutes to keypunch on an IBM 029 keypunch. In addition to a keypunch for preparing documents for publication, a PL/I compiler, 3 tape drives, a card reader nd line printer will be required. In addition all of the index programs require access to the SORT/MERGE routines. The access to SORT/MERGE will be more efficient if disk storage is available; however if sufficient tape drives are available, the sorting can be handled on tapes.

Storage of Data Pertaining to Clearinghouse Documents

Storage and maintenance of data is a first problem when preparing to create a publication. What format should be used for the data? What storage device should be used? These are only two of the questions to be answered. One alternative is storage on disk providing a direct access system which would be very efficient. The problem with this approach is the storage costs, which are relatively high, thus magnetic tape storage was advisable. Data must be prepared for input on available equipment and, in this case, it amounted to a choice between paper tape and punched cards.

Because of available equipment, paper tape would have to be converted to punched cards as an intermediate step to magnetic tape, thus initial preparation of cards was a logical decision. If the hardware available had allowed direct conversion from paper tape to magnetic tape, the choice to use paper tape would have been feasible. Temporary retention of data cards affords security for the data; but, after printing of the publication, these cards may be destroyed.

The bibliographic data in ERIC resumes varies both in length and in types of data available. Originally, the Clearinghouse format was variable and differed from the format being used by Central ERIC . This variable format was the basis for card preparation and magnetic tape file creation. The program to convert the card data to magnetic tape is referred to as the MASTER program.

The MASTER program reads in each card for a particular document and analyzes the card for syntax. This program pulls off all card images related to a document by using the accession number. The accession number is in the first ten positions of the record. Then the program searches the card images to discover what data is applicable and what is or is not included. The program then sets up one record on the MASTER tape for the document. The length of the record varies due to the varying material, and there is a record for every document in the quarterly abstract series (ie., AIM and ARM). This record is set up so that all bibliographic data is listed first in a special order, followed by a section of numbers that gives the lengths of the different data. A zero length implies that a data piece is missing from the record. The lengths and zeroes allow access to any particular piece of bibliographic information within this large record, and the zero lengths indicate certain bibliographic information is missing. The order and format

of the data in each record is as follows: (Asterisks indicate material which does not always appear.)

Bibliographic Record Format

	Piece of Data	Length (in bytes)
1.	Document Identification Number (VT Number)	10
2.	Code Number	17
3.	Accession Number (ED Number)	20
4.	Whether it is an AIM or ARM publication and the current season	3
5	Current Year	4
6.	Page number on which the resume appears in the front section of publication	4
7.	*$Author_1$, . . . $Author_n$, $n \leq 2$	≤ 132
8.	Title	≤ 396
9.	*Corporate $Author_1$, . . . $Author_n$, $n \leq 5$	≤ 660
10.	*Sponsoring $Agency_1$, . . . Sp. $Agency_n$, $n \leq 3$	≤ 396
11.	Price, Contract Number, Grant Number, and Program Area Number	≤ 132
12.	*Report Number, Bureau Number	≤ 132
13.	*Availability	≤ 264
14.	*Journal Citation	≤ 132
15.	Publication Date	7
16.	The number of pages in the report	≤ 16

17.	*Descriptive Note	≤247
18.	Descriptors	660
19.	*Identifiers	≤396
20.	// (separator)	2
21.	Number of authors in Section 7, 0≤n≤2	4
22.	*Length of each author in Section 7 (If n=0, then there is no Section 22.)	4*n
23.	Length of the title	4
24.	Number of corporate authors in Section 9 0≤n≤5	4
25.	*Length of each corporate author in Section 9 (If n=0, there is no Section 25.)	4*n
26.	Number of sponsoring agencies in Section 10 0≤n≤3	4
27.	*Length of each sponsoring agency in Section 10 (If n=0, there is no Section 27.)	4*n
28.	Length of the Price	4
29.	Length of the Contract Number	44
30.	Length of the Grant Number	4
31.	Length of the PA Number	4
32.	Length of the Report Number	4
33.	Length of the Bureau Number	4
34.	Length of the Availability	4
35.	Length of the Journal Citation	4
36.	Length of the Publication Date	4
37.	Length of the Page Count	4
38.	Lenth of the Descriptive Note	4
39.	Length of the Descriptors	4
40.	Length of the Identifiers	4

The maximum length of such a record is 3,732 bytes.

Many of the Clearinghouse products of the tape to be produced by this package have been announced in RIE at an earlier date. In the case of these publications, the data is already in machine readable form. The problem of needing a separate format for the publication tapes is easily solved by a program to convert the tapes to the desired Clearinghouse format. Since only bibliographic data is needed for index sections of the publications, the program designed to convert the RIE magnetic tape data create two new magnetic tapes: one tape for bibliographic data and the second one for abstracts for the documents.

In order to make the publications more useful than a simple bibliography, it is necessary to add abstracts to the finished product. Since the abstract is only used for the resume section of the publications, it is more efficient to have the abstracts on a separate tape.

The format used for storing abstract data is very simple: 10 bytes for the Clearinghouse accession number, 2 bytes to identify the subject area, and 1,988 bytes for the abstract. The 2-byte code used in the abstract data appears in a fixed position in the bibliographic data. The Clearinghouse accession number and the code is both bibliographic and abstract data; thus, the documents can be sorted into several different orders and still allow a program to read both tapes and find corresponding document records.

Production of Clearinghouse Information Products

The first step is to determine which documents are to appear in the Clearinghouse publication. After the documents have been selected, the location of the bibliographic and abstract data must be determined. If the information has appeared in RIE and the quarterly update of ERIC tapes, the document information will not need to be keypunched.

Once the document information is located, bibliographic and abstract tapes need to be created. Once these magnetic tapes have been created, a resume section containing abstracts and bibliographic data can be printed. Then, author, subject, supporting service, and title (or document number) indexes may be printed.

After the resume and indexes are proofed, the camera-ready printout can be prepared and then processed by a printing contractor. This involves reduction of the size and printing of the publication.

Retrieval of Clearinghouse Documents

The bibliographic and abstract tapes can be combined into the ERIC tape format to permit use of QUERY. QUERY allows the capability to search the data base.

II

DESCRIPTION OF COMPUTER PROGRAM OPERATING PROCEDURES, CAPABILITY, AND UTILIZATION IN ERIC CLEARINGHOUSES

Procedures for creation of document number, author, supporting services, subject, and document number conversion indexes are described in the following sections. Further detail on specific programs is provided in the appendixes referred to in the text. Also described are edit programs for bibliographic records, annual document number index data, and all other annual index data. Use of the QUERY programs on Clearinghouse magnetic tapes is explained.

Indexes

Development of the indexes is dependent on the existence of the bibliographic and abstract magnetic tapes that can be created following procedures described in preceding sections of the report. Control data for all indexes are listed in Appendix C. Listings are in Appendix E and flowcharts are in Appendix F.

The Document Number Index

In the document number index, Clearinghouse accession number, ED number, resume section page number, and title for each document are listed. The order of the index is by ascending accession number. In order to produce the document number index, the MASTER file containing the bibliographic information is read by the document number index program. The information needed for the document number index is pulled from the bibliographic tape. Since the year of publication will be needed in the annual index, this is also pulled from the bibliographic tape. The information for each document is stored in its own record on a separate device (i.e., disk or tape). The subset file which is thus created is then sorted by accession number. The records in the subset file consist of:

4. Length of title (4 bytes)
5. Title (maximum of 396 bytes)
6. Issue (3 bytes)
7. Year (4 bytes)

Total record length is $\leq$ 441 bytes

After these records have been sorted by accession number, the document number index for the quarterly issue is created. This index is first printed and proofed. It is then placed on magnetic tape, and the data on this tape is printed on camera-ready paper. After the index for quarterly issue has been prepared, the annual index can be prepared from this and data sets from other quarterly issues. All of the information from the indexes in the quarterly issue is not used in the annual index. Thus, each record is shortened to contain:

1. Accession number (6 bytes)
2. Page number (4 bytes)
3. Issue (3 bytes)
4. Year (2 bytes)
5. Three blanks to increase length to 18 bytes.

The subset file for the annual index is stored on magnetic tape and this data set is sorted by ascending accession number. The annual document number index contains the shortened records from the document number indexes for the issues which have been published during the year.

Supporting Services Index

To produce the supporting services index, Clearinghouse accession number, ED number, resume section page number, title, and part of the special code (i.e., form, field, and category) is needed. The section of the code needed indicates the field for the document. This code is the 3 characters in the 20-22 column of the code card. For example, in vocational and technical education, these fields might be: agricultural education, business education, distributive education, home economics education, health occupations education, industrial arts education, manpower economics, occupational psychology, occupational sociology, trade and industrial education, and general technical and vocational

education. The program is presently set for the fields in vocational and technical education. However, these can easily be adjusted to suit any supportive service required or other categories. The supporting service is not repeated for each document. It is listed once and all the documents having that sub-code are listed under the title. The documents are listed in clearinghouse accession number ascending order under the appropriate supporting service. For each bibliographic record, sub-codes, clearinghouse accession number, ED number, resume section page number, and title are needed for the indexes in the quarterly issues. The publication date will be needed for the annual index. This information is stored on a subset file containing:

1. Sub-code (3 bytes)
2. Clearinghouse accession number (19 bytes)
3. ED number (20 bytes)
4. Publication date (7 bytes)
5. Title (maximum of 396 bytes)

Total record length is $\leq$ 436 bytes.

After the quarterly issue is prepared, the subset files can be shortened further for the annual index. The annual supporting services index, sub-code, publication year, and clearinghouse accession number are the only information needed. The format for the annual supporting services subset file is:

1. Sub-code (3 bytes)
2. Publication year (2 bytes)
3. Clearinghouse accession number (6 bytes)
4. Seven blanks at the end for padding to make record length 18 bytes.

Subject Index

The subject index is made up from starred descriptors and identifiers. The bibliographic tape format is

$$keyword_1; \ldots keyword_n; \text{ etc.}$$

Some keywords are starred. These are the terms to be used as index terms. For the subject index, it is necessary to extract all starred identifiers and descriptors, Clearinghouse accession number, ED number, resume section page number, and title for the quarterly subject index, plus the publication year

for the annual subject index. For each starred keyword in the record, a record is written on a disk subset file. There are approximately five starred keywords for each record. The records on disk subset files will look as follows:

1. Starred keyword (200 bytes)
2. VT number (10 bytes)
3. ED number (20 bytes)
4. Page number (4 bytes)
5. Publication date (7 bytes)
6. Title ($\leq$ 396 bytes)

Total record length is $\leq$ 637 bytes.

In order to sort according to the starred keywords, the keyword field in the record must be of a fixed length. However, the keywords vary in length, so a maximum keyword length of 200 bytes was selected. When a keyword is not 200 bytes long, it is left justified and then padded with blanks on the right. These records are first sorted by alphabetical order of keywords and then by accession number in ascending order. The subject index is printed and proofed, then copied onto tape to be put on camera-ready paper. Once the subject index has been produced for the quarterly issue, the annual subject subset file can be written. For the annual index, the records will contain:

1. Starred keyword (200 bytes)
2. Publication year (2 bytes)
3. VT number (6 bytes)

Record length is 208 bytes.

Author Index

Each bibliographic record can have personal authors or corporate authors or perhaps neither of the two. The author index uses the author name, accession number, resume section page number, and title for the author index in the quarterly issue, and the publication date for the annual author index. The author index is sorted by authors in alphabetical order. Each author name is listed once in the index and under it is listed all documents written by the author in Clearinghouse accession number ascending order. The record for each author is formed in the author subset file. The records consist of:

1. Author name (132 bytes)
2. Clearinghouse accession number (10 bytes)
3. ED number (20 bytes)
4. Page number (4 bytes)
5. Year (2 bytes)
6. Length of title (3 bytes)
7. Title (maximum of 396 bytes)

Total record length is 567 bytes.

Just as in the subject index, the author name is padded with blanks on the right to enable the sort, the subset file records are sorted by author name in alphabetical order and then by Clearinghouse accession number in ascending order. Once this file has been sorted, the index is then printed: first on lined paper for proofing, and then on tape for preparing the camera-ready copy.

As the index is being formatted for the quarterly issue, the annual author subset file is being formed. The format of the annual records is:

1. Author name (132 bytes)
2. Publication year (2 bytes)
3. Clearinghouse accession number (6 bytes)

Total record length is 140 bytes.

As in the other indexes, at the end of the year, the quarterly subset files are combined to form the annual index.

Conversion of Document Number Index

The conversion of document number index is formed from the punched cards prepared by the resume print program. The cards punched by the resume program contain the Clearinghouse accession number, ED number, and resume section page number. The information necessary for this index is the Clearinghouse accession number and ED number. Since these decks are rather small, the deck for each index can be saved and the information from one quarter to the next can be combined into one annual index.

The index itself consists of two parts: the conversion of Clearinghouse accession numbers to ED or MP numbers, and the conversion of ED or MP numbers to Clearinghouse accession numbers. Both of these sections may be cumulative.

The annual conversion of document number index may be formed by combining the Clearinghouse accession numbers to ED numbers sections of the indexes from each quarterly issue. The annual index contains only the Clearinghouse to ED number conversion.

Edit Programs

Since errors may be made in preparing the data base, two points have been identified as requiring an edit capability. The first is the MASTER tape--if this tape is to serve the key role it was designed for, then it must be free of errors. The second point is the annual subset files. With the possibility of an error being missed until the annual is to be produced, an edit capability should be available, avoiding the need to rerun a quarter's publication for the sake of an error in the subset file. The MASTER tape is eventually corrected; however, the second edit allows publication schedules to proceed.

Master Edit Program

Some resumes will be found to contain errors during proofing of the printout. The first step in editing is to read the list of Clearinghouse accession numbers on the tape to identify the documents needing correction. The remaining documents are copied onto a new tape. Incorrect resumes are corrected by punching new data cards to replace those in error. The resumes of corrected documents are added to the end of the new tape. The MASTER edit program also allows the deletion of documents in order to obtain the desired paging in the resume section.

Annual Subset Edit Program for Author, Subject, and Supporting Services Indexes

One program will handle the editing for the author, subject, and supporting services indexes. There are three phases to this program: (1) replace any records which have errors in them, (2) delete any excess records, and (3) add

any records which have been skipped. For each of the phases, the keyword, publication date, and Clearinghouse accession number are read. The keyword is either the subject, the author, or the code for the supporting service. In the case of the replacement phase, both the data for the bad document and the data for the correct document are needed.

Annual Subset Edit Program for Document Number Index

A separate program is needed to edit the document number index. The program is similar to the edit program for the other subset files; however, due to the change in content and number of fields, a separate program is needed.

Use of QUERY Program

QUERY, primarily designed for retrieving information from the tapes in the ERIC format, is also useful for retrieving information from Clearinghouse collections provided that they are also in ERIC format.

The first use of QUERY in this software package is for retrieving information from RIE for the Clearinghouse publications. Many documents that the Clearinghouse may want to use will already have appeared in RIE. Once these records have been extracted, the ERIC format is converted to the Clearinghouse bibliographic format and abstract format by use of a program listed in Appendix E and flowcharted in Appendix F. The bibliographic tape then can be used with the index programs to prepare the index sections of the publication. The bibliographic tape and abstract tape are then used with the resume print program to prepare the resume section for publication.

QUERY can also be used for obtaining documents pertaining to a particular topic. These documents could then be processed through the programs, and publications containing resume sections and indexes could be prepared. When the Clearinghouse has published its own documents, the question arises: How do they search these documents? When the document base becomes large enough that computer applications are feasible, the software package contains programs

to convert the document base to the ERIC format (Appendix E and F). With the document in the ERIC format, the tape can be searched with QUERY, just as the original ERIC tapes could be searched.

III

SUMMARY AND RECOMMENDATIONS

Summary of the Project

The purpose of this project was to develop a software package for the production of Clearinghouse abstract journal publications and creation of magnetic tape files for use in document retrieval. The programs in the package are written in PL/I for use on IBM 360 or other computers with a PL/I compiler. Also needed are keypunch machines, three tape drives, card reader, and line printer. A SORT/MERGE program is needed to supplement the routines prepared especially for this software package.

Programs in the software package facilitate preparation of camera-ready copy of resume sections, and subject, author, supporting service, document number, and document number conversion indexes. Search tapes may also be created for use in computer searching of the Clearinghouse collection or repackaging of the document base in other forms.

The procedures require that document data be stored on magnetic tapes. Input to the bibliographic and abstract tapes is by punched data cards for non-RIE documents and through format conversion of RIE documents identified in a QUERY search of the ERIC tapes. Data on bibliographic and abstract tapes are merged into a resume section which is formatted and printed. The bibliographic tape is used to format and print subject, author, supporting services, and document number indexes. Cards created as an outcome of the resume section print program are used to format and print the document number conversion index. A search tape is created by merging bibliographic and abstract tapes.

The utility of the complete software package has been proven in production of the Fall 1970 issue of Abstracts of Research and Related Materials in Vocational and Technical Education (ARM). Portions of the package have been used in

production of Spring, Summer, and Fall (1970) issues of Abstracts of Instructional Materials in Vocational and Technical Education (AIM); the Spring and Summer (1970) issues of ARM; and the 1969-70 Annual Index.

The software package has been documented and is suitable for installation in other Clearinghouses. Feasibility has been checked with five other Clearinghouses (i.e., Counseling and Personnel Services, Educational Media and Technology, Reading, Rural Education and Small Schools, Science and Mathematics Education).

Recommendations

It is recommended that ERIC Clearinghouses be advised of availability of the software package. This announcement should encourage installation by any Clearinghouse which has a document collection sufficiently large to warrant computer retrieval capacity and by Clearinghouses which produce bibliographies and/or abstract journals from RIE and Clearinghouse document collections. This will require duplication of software documentation appended to this report.

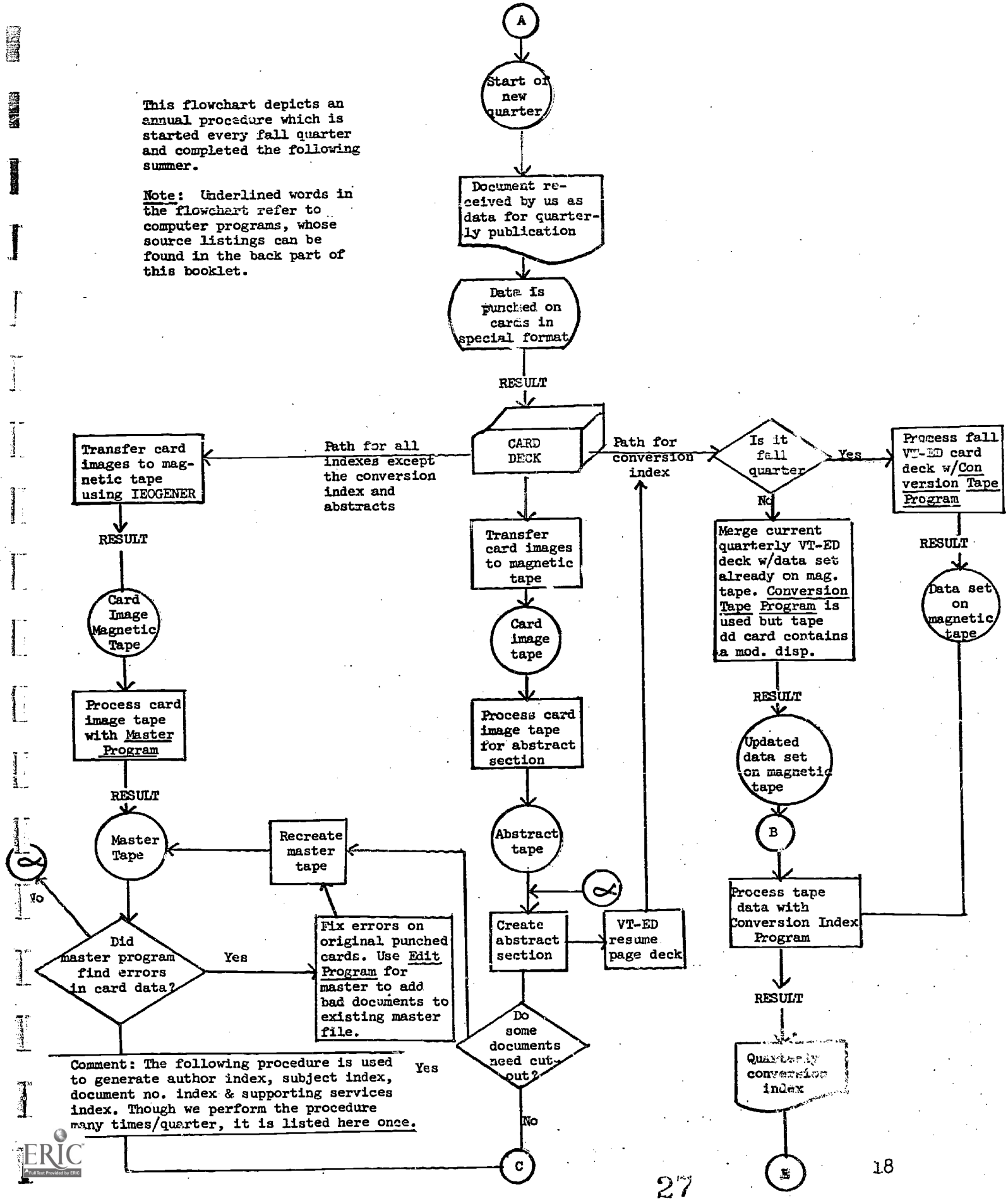
It is also recommended that provisions be made for training sessions, visitation, or consultation necessary to successful installation and utilization of the software package.

It is further recommended that additional procedures be developed so that the bibliographic tape be created when a document is first accessioned by the Clearinghouse. Additional routines could be developed which would permit duplicate checking, preparation of the Acquisition Data List, source checking, and production of control file cards.

Flowchart of the System

(The System Is the Same for AIM and ARM)

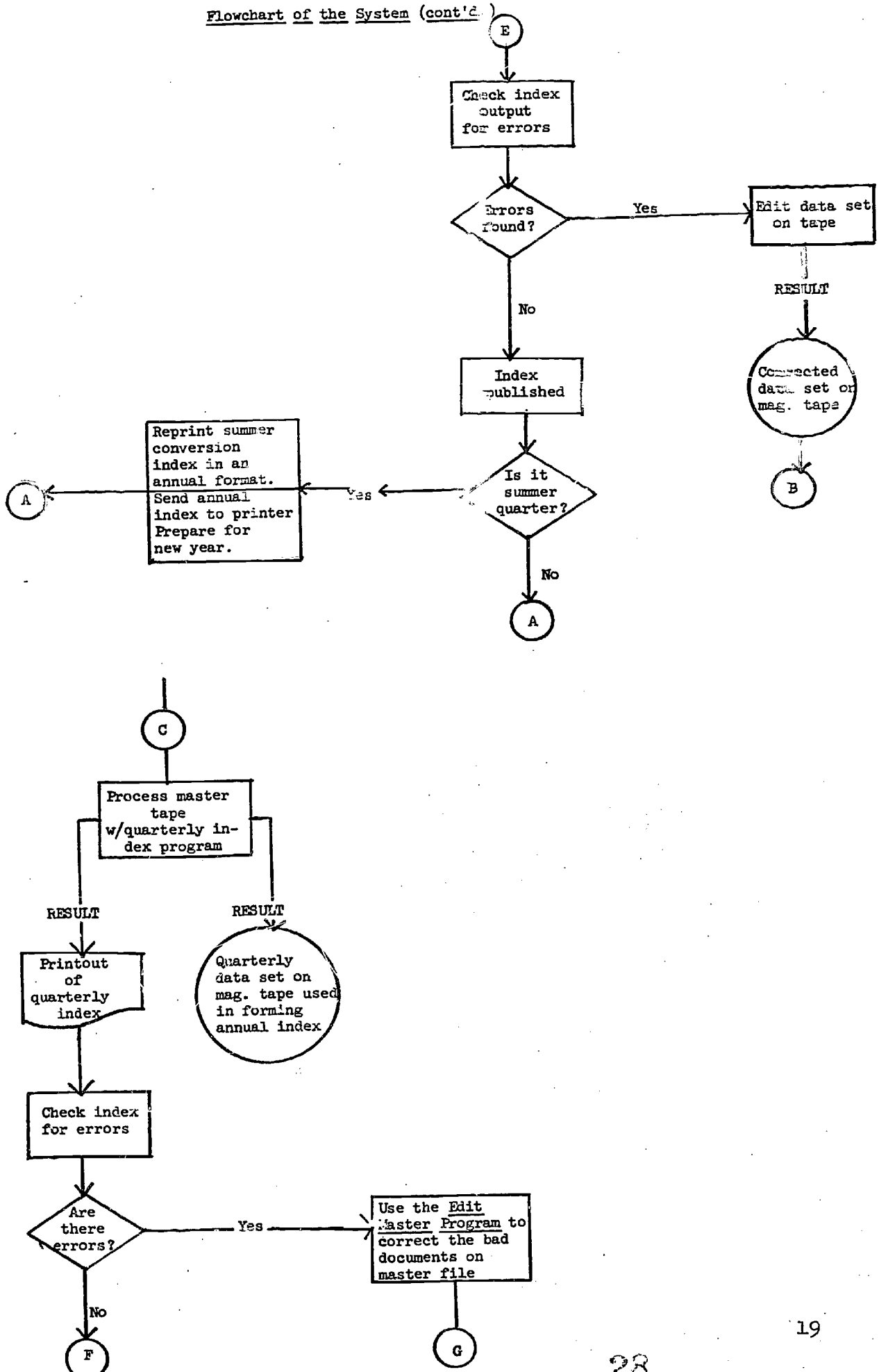
Figure 2



This flowchart depicts an annual procedure which is started every fall quarter and completed the following summer.

Note: Underlined words in the flowchart refer to computer programs, whose source listings can be found in the back part of this booklet.

Flowchart of the System (cont'd.)



Flowchart of the System (cont'd.)

Figure 2

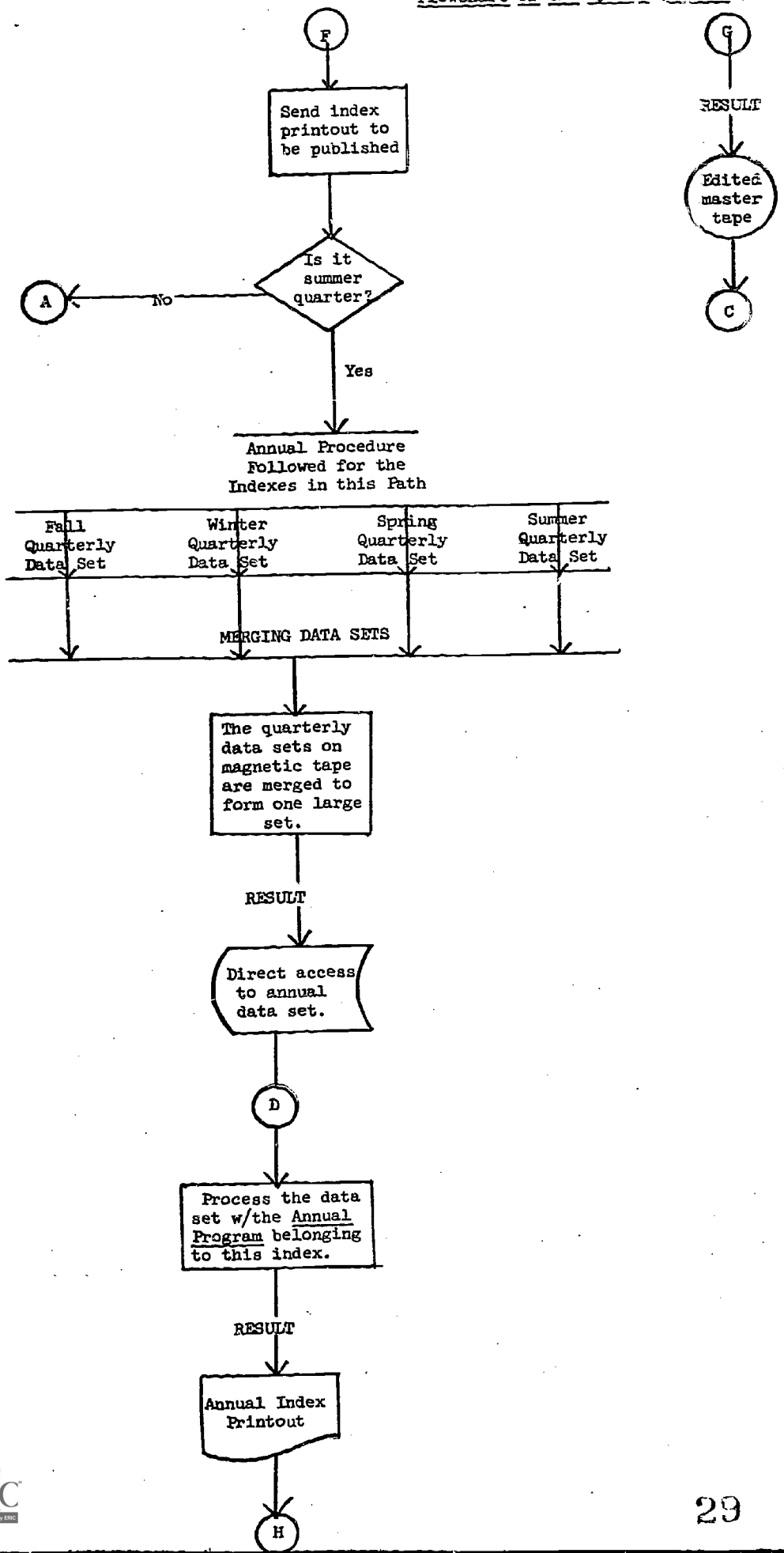
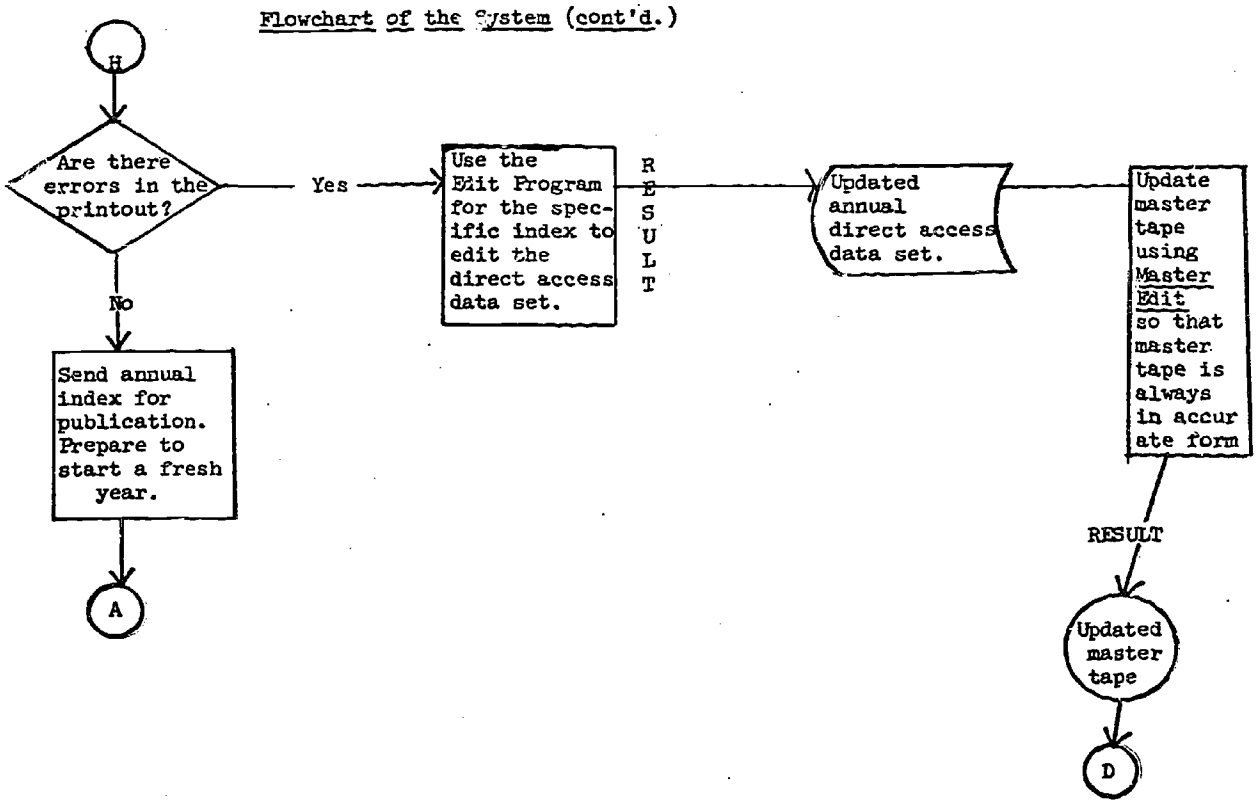


Figure 2

Flowchart of the System (cont'd.)



APPENDICES

Number		Page
A.	KEYPUNCHING PROCESS	2
B.	JOB CONTROL LANGUAGE	7
C.	CONTROL CARDS	36
D.	ERROR MESSAGES	53
E.	PROGRAM LISTINGS	58
F.	PROGRAM AND SYSTEM FLOWCHARTS	210
G.	PAGE SAMPLES	264

APPENDIX A

KEYPUNCHING PROCESS

Appendix A

The Process of Keypunching for Clearinghouse Documents

The Punching Process

The VT number uses the first 10 columns of the punch card, followed by the line and card number on the next 4 columns, with the data starting on column 15.

Example: VT 000 0000011Data.....

The bibliographic cards have a zero added before the line number and a 1 or 2 after the line number that designates the card number. This gives the card a 4 number digit. For example: line 20 goes to 0201 - 0202, etc., depending on how many cards are needed for the title. If there are more than two cards needed, the number goes to 0211 - 0212, 0221 - 0222, keeping them in the 20 class but building up the series. The 100 and 200 classes have only the card number added, to make the fourth digit. For example: 2001 - 2002.

A program card is used to control the keypunch and automatically reproduces the VT number from card to card. At the end of each number class of information, an end-of-line mark is used to show the end. This is the horizontal bar on the keypunch. Example: If class number 30 went onto only one card it would be

VT 000 0000301Data.....|

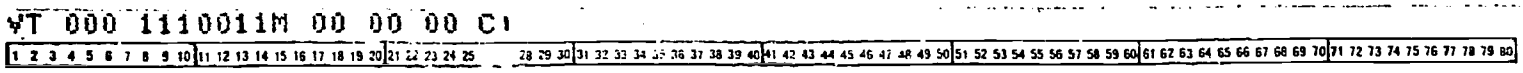
or if on two cards, would be

VT 000 0000301Data.....
VT 000 0000302.....|

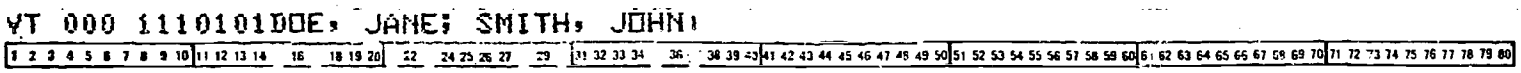
Bibliographic Section

00

01 These two lines are combined into the VT code card. Two of these cards are needed, one for the bib section and one for the abstract section, as these cards are kept separate in the files. The end-of-line mark is also used at the end of this card. Example:



10 This card contains the Personal Author data and uses the end-of-line mark. If more than one author appears, they are separated with a semi-colon. Example:



20 This card contains the Title of Document and also has special editing marks used by the computer. These are short title marks showing the beginning and end of the title wanted for the Subject Index so they are placed at various places within the title cards, and the end-of-line mark is used. The title marks are the "less than" mark at the beginning and the "more than" mark at the end. Example:

```
VT 000 1110201<WALLPAPER AND PAINT>.. WORKBOOK.¹
```

30
31
32 These cards are the Corporate Author data cards and uses the end-of-line
33 mark. These names must have a period before the end-of-line mark. Example:
34

```
VT 000 1110301OHIO STATE UNIV., COLUMBUS. DISTRIBUTIVE EDUCATION SERVICE.¹
```

```
VT 000 1110311OHIO OHIO STATE UNIV., COLUMBUS. DISTRIBUTIVE EDUCATION MATERIALS
```

```
VT 000 1110312LAB.¹
```

40
41 These cards are the Sponsoring Agency data cards and uses the end-of-line
42 mark. Example:

```
VT 000 1110401OHIO STATE DEPT. OF EDUCATION, COLUMBUS¹
```

50 The card contains the price line, contract number, grant number, and PA number. There are also special editing marks used in this class and the end-of-line mark. The special edit marks are the tab or ¬ on the punch. These are placed after the price, after the contract number, after the grant number and end-of-line after the PA number. For example: Price ¬ Contract ¬ Grant ¬ PA /. If there are no contract, grant or PA, these 3 marks must still be shown. Example:

```
VT 000 1110501EDRS PRICE MF-$0.25  HC-$1.50¬¬0E12345¬08¹
```

```
VT 000 1110501EDRS PRICE MF-$0.25  HC-$1.50¬¬¬¹
```

60 This card contains the Report number and the Bureau number. There is also a tab used if one or the other is not available, and the end-of-line mark. For example: 0601Report→! or 0601→Bureau! or if both are present the tab is used to separate them. Example:

```
VT 000 1110601PUB-123→!
```

70 This card contains the availability of the document and the end-of-line mark. Example:

```
VT 000 1110701THE CENTER FOR VOCATIONAL AND TECHNICAL EDUCATION, THE OHIO STATE
VT 000 1110702UNIVERSITY, 1900 KENNY ROAD, COLUMBUS, OHIO 43210 ($2.00).!
```

80 The Journal Citation is on this card. There is also an end-of-line mark. Example:

```
VT 000 1110801AMERICAN VOCATIONAL JOURNAL; V49 N10 P49-51 DECEMBER 1970!
```

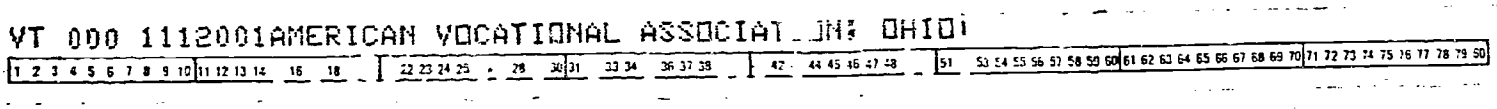
90 The date, pages and any descriptive note are on this line and editing marks are also used. The first seven spaces starting with column 15 are used for the date, a tab to separate from the pages, the page number and another tab to separate from the descriptive note and the end-of-line mark. Example: 0901Date→Page→! or 0901Date→Page→Descriptive note!. The 2 tabs must always appear. Example:

```
VT 000 1110901310V69→112P.→PAPER PRESENTED AT AMERICAN VOCATIONAL ASSOCIATION C
VT 000 1110902ONFERENCE!
```

100 This class contains the Descriptors and uses the end-of-line mark. The descriptors are separated with a semi-colon and the major descriptors are designated with an asterisk. Example:

```
VT 000 1111001*TEXTBOOKS; *STUDY GUIDES; HIGH SCHOOLS; POST SECONDARY EDUCATION;
VT 000 1111002 *INDIVIDUAL INSTRUCTION!
```

200 This is the Identifier class and uses the end-of-line mark. These are also separated with a semi-colon. Example:



Abstract Section

The abstract section is handled the same way as the bibliographic section. The cards carry the VT number and line class starts with 300, and carries the 1 or 2 card number. Example: 3001 - 3002, 3011 - 3012, etc. No zero is added before the line number. These cards also have a second run through the keypunch so the zone punch can be added to all lower case letters.

Adding the Lower Case Punch

Once the cards are punched correctly the first time, any line having lower case is run back through the punch a second time adding the zone punches that will cause the letters to be dropped to lower case on the main computer run for proofing and publication. This data is designated in lower case on white-forms. The zone punches are the numerical "zero" zone punch, which punches in the zero row and prints a zero over the letters A through I; the 12 punch which punches in the 12 row and prints an ampersand or "&" over letters J through R; and the 11 punch which punches in the 11 row and prints a dash over letters S through Z. The numeric shift must be held down on these punches.

After a document is finished, and end-of-document card is placed at the end of the bib section and the abstract to designate to the computer that that document is finished. The EOD card has a tab (→) in column 1, with the rest of the card blank.

APPENDIX B

JOB CONTROL LANGUAGE

	page
RIE TO AIM/ARM	8
MASTER	10
ABSTRACT	11
MASTER SORT	12
RESUME PRINT	13
MASTER EDIT	14
SUBJECT QUARTERLY	15
AUTHOR QUARTERLY	17
SUPPORTING SERVICES QUARTERLY	19
DOCUMENT NUMBER QUARTERLY	21
CONVERSION OF DOCUMENT NUMBER	23
SUBJECT ANNUAL	25
AUTHOR ANNUAL	27
SUPPORTING SERVICES ANNUAL	29
DOCUMENT NUMBER ANNUAL	32
EDIT OF SUBJECT, AUTHOR, AND SUPPORTING SERVICES	33
DOCUMENT NUMBER EDIT	34
RIE FORMAT	35

NOTE: J.C.L. is for either object module or load module. Both forms are illustrated. If need to recompile, check with a programmer.

RIE TO AIM/ARM

1. Object Module

```
//FARM EXEC PGM=LOADER,PARM='LET',TIME=2
//SYSLIB DD DSN=SYS1.PL1LIB,DISP=SHR
//SYSLOUT DD SYSOUT=A
//SYSLIN DD *
//SYSPRINT DD SYSOUT=A
//SYSIN DD *
/*
//VTSIN DD *
/*
//WORK DD UNIT=SYSDA,SPACE=(CYL,(1,1)),DISP=(,PASS),DCB=(RECFM=V,
// LRECL=3736,BLKSIZE=3740)
//VTABS DD UNIT=SYSDA,SPA  =(CYL,(1,1)),DCB=(RECFM=V,LRECL=2000,
// BLKSIZE=2004),DISP=(,PASS)
//RIE DD UNIT=2400,VOL=SER=ERIC25,DISP=OLD,DSN=VTERIC,DCB=(RECFM=V,
// LRECL=3617,BLKSIZE=3625)
```

WORK is the data set for the MASTER records. The normal storage media will be magnetic tape.

VTABS is the data set for the abstract records.

RIE is the ddname for the data set containing the original records in RIE format.

2. Load Module

```
//JOBLIB DD DSN=ERICLIB,DISP=SHR
//RIETOUT EXEC PGM=VTBIB,TIME=2
//SYSPRINT DD SYSOUT=A
//SYSIN DD *
/*
//VTSIN DD *
 /*
```

RIE TO AIM/ARM (cont'd.)

Load Module (cont'd.)

```
//WORK DD UNIT=2400,VOL=SER=ERICB,DISP=(,PASS),
// DCB=(RECFM=V,LRECL=3736,BLKSIZE=3740)
//VTABS DD UNIT=2400,VOL=SER=ERICA,DISP=(,PASS),
// DCB=(RECFM=V,LRECL=2000,BLKSIZE=2004)
//RIE DD UNIT=2400,VOL=SER=ERIC25,DISP=OLD,
// DSN=VTERIC,DCB=(RECFM=V,LRECL=3617,BLKSIZE=3625)
```

MASTER

1. Object Module

```
//FARM EXEC PGM=LOADER,PARM='LET',TIME=2
//SYSLIB DD DSN=SYS1.PL1LIB,DISP=SHR
//SYSLOUT DD SYSOUT=A
//SYSLIN DD *   MASTER obj. follows
          obj. deck
/*
//SYSPRINT DD SYSOUT=A
//ERRE DD SYSOUT=A   Print error messages
//MASTER DD UNIT=2400,VOL=SER=ERIC04,LABEL=(5,SL),DSN=FARM,
// DCB=(RECFM=V,LRECL=3736,BLKSIZE=3740),DISP=(NEW,KEEP)
//SYSIN DD *  Control cards and VT-ED deck
        see MASTER listing for card information
/*
//CARDS DD *   bibliographic cards
/*
```

The card //CARDS DD * may be replaced with a card referring to a tape if cards are on tape, e.g.:

```
//CARDS DD UNIT=2400,VOL=SER=ERIC18,LABEL=(10,SL),DISP=(OLD,KEEP)
// DSN=CFAIM T,DCB=(RECFM=FB,LRECL=80,BLKSIZE=3200)
```

2. Load Module

```
//JOBLIB DD DSN=ERICLIB,UNIT=2314,VOL=SER=CL0091,DISP=SHR
//FARM EXEC PGM=MASTER,TIME=2
//SYSPRINT DD SYSOUT=A
//ERRE DD SYSOUT=A
//MASTER DD  definition of tape to hold records
//SYSIN DD *    control cards
/*
//CARDS DD   definition of tape holding cards or * if card input
/*
```

ABSTRACT

1. Object Module

```
//FARM EXEC PGM=LOADER,PARM='LET',TIME=2
//SYSLIB DD DSN=SYS1.PL1LIB,DISP=SHR
//SYSLOUT DD SYSOUT=A
//SYSLIN DD *  object deck for abstract
//SYSPRINT DD SYSOUT=A
//SORTIN DD UNIT=2400,VOL=SER=ERIC27,DSN=FLRMABS,DISP=(,PASS),
// DCB=(RECFM=V,LRECL=2000,BLKSIZE=2004)
//SYSIN DD *   control data
/*
//CARDS DD *   abstract data
/*
```

2. Load Module

```
//JOBLIB DD DSN=ERICLIB,DISP=SHR
//ABS EXEC PGM=ABSGEN,TIME=2
//SYSPRINT DD SYSOUT=A
//SORTIN DD UNIT=2400,VOL=SER=ERIC27,DSN=FLRABS,DISP=(,PASS),
// DCB=(RECFM=V,LRECL=2000,BLKSIZE=2004)
//SYSIN DD *   control data
/*
//CARDS DD *   abstract data
/*
```

MASTER AND ABSTRACT SORT

```
//JOBLIB DD DSN=SYS1.JOBLIB,DISP=SHR
//SORT EXEC PGM=IERRCOOO,TIME=1
//SYSOUT DD SYSOUT=A
//SORTLIB DD DSN=SYS1.SORTLIB,DISP=SHR
//SORTIN DD UNIT=2400,VOL=SER=ERIC04,DSN=SUAIM,LABEL=(4,SL),DISP=OLD,
//DCB=(RECFM=V,LRECL=3736,BLKSIZE=3740)
//SORTOUT DD UNIT=SYSDA,SPACE=(CYL,(1,1)),DISP=(,PASS),
//DCB=(REDFM=V,LRECL=3736,FLKSIZE=3740)
//SORTWK01 DD UNIT=SYSDA,SPACE=(TRK,20,,CONTIG)
//SORTWK02 DD UNIT=SYSDA,SPACE=(TRK,20,,CONTIG)
//SORTWK03 DD UNIT=SYSDA,SPACE=(TRK,20,,CONTIG)
//SORTWK04 DD UNIT=SYSDA,SPACE=(TRK,20,,CONTIG)
//SORTWK05 DD UNIT=SYSDA,SPACE=(TRK,20,,CONTIG)
//SORTWK06 DD UNIT=SYSDA,SPACE=(TRK,20,,CONTIG)
//SYSIN DD *
```

The control cards in the SYSIN data set are of the form

```
SORT FIELDS=(x,2,CH,A,5,10,CH,A),SIZE=E500
END
```

The number used in place of x depends on the information being sorted.

Abstract data	use 17
ARM data	use 23
AIM data	use 21

RESUME PRINT

1. Object Module

```
//FARM EXEC PGM=LOADER,PARM='LET',TIME=2
//SYSLIB DD DSN=SYS1.PL1LIB,DISP=SHR
//SYSLOUT DD SYSOUT=A
//SYSLIN DD *
//SYSPRINT DD SYSOUT=A
//OUT DD SYSOUT=A,DCB=(RECFM=VA,LRECL=137,GLKSIZE=141)
//NEWMAS DD UNIT=2400,DISP=NEW,DCB=(RECFM=V,LRECL=3736,BLKSIZE=3740),
                                                    VOL=SER=ERIC
//MASTER DD DSN=*.SORT.SORTOUT,DISP=(OLD,PASS)
//PUNCH DD DUMMY,DCB=(RECFM=F,LRECL=80,BLKSIZE=80)
//SORTOUT DD UNIT=2400,VOL=SER=ERIC21,DSN=SUAMABS,DCB=(REDFM=V,
// LRECL=2000,BLKSIZE=2004),DISP=OLD
//SYSIN DD *
```

NEWMAS is the MASTER data set with the resume pages added.

MASTER is the bibliographic data without page numbers

PUNCH is for the VT-ED resume page decks.

SORTOUT is the abstract data to be read.

2. Load Module

```
//JOBLIB DD DSN=ERICLIB,DISP=SHR
//ABSPRT EXEC PGM=ABSPRINT,TIME=2
//SYSPRINT DD SYSOUT=A
//OUT DD UNIT=2400,DISP=NEW,DCB=(RECFM=V,LRECL=3736,VLKSIZE=3740),
                                                    VOL=SER=ERIC
//MASTER DD DSN=*.SORT.SORTOUT,DISP=(OLD,PASS)
//PUNCH DD DUMMY, DCB=(RECFM=F,LRECL=80,BLKSIZE=80)
//SORTOUT DD UNIT=2400,VOL=SER=ERIC21,DSN=SUAMABS,DCB=(RECFM=V,
// LRECL=2000,BLKSIZE=2004),DISP=OLD
//SYSIN DD *
```

MASTER EDIT

1. Object Module

```
//FARM EXEC PGM=LOADER,PARM='LET',TIME=2
//SYSLIB DD DSN=SYS1.PL1LIB,DISP=SHR
//SYSLOUT DD SYSOUT=A
//SYSLIN DD *   object deck for MASTER edit
/*
//SYSPRINT DD SYSOUT=A
//ERRE DD SYSOUT=A    print error messages
//SYSIN DD *   control cards and VT-ED deck
/*
//CARDS DD *   bibliographic information for edits
/*
//MAST1 DD UNIT=2400,DISP=(OLD,KEEP),LABEL=(4,SL),DSN=WNTAIM,
// DCB=(RECFM=V,LRECL=3736,BLKSIZE=3740),VOL=SER=ERIC02
//MASTER DD UNIT=2400,DISP=(NEW,KEEP),LABEL=(1,SL)DSN=WNTAIM,
// DCB=(RECFM=V,LRECL=3736,BLKSIZE=3740),VOL=SER=ERIC04
```

MAST 1 is the data set to be edited.

MASTER will contain the edited bibliographic file.

2. Load Module

```
//JOBLIB DD DSN=ERICLIB,DISP=SHR
//MASED EXEC PGM=MASEDT,TIME=2
//SYSPRINT DD SYSOUT=A
//ERRE DD SYSOUT=A     print error messages
//SYSIN DD *   control cards and VT-ED deck
/*
//CARDS DD *   bibliographic information for eidts
/*
//MAST1 DD UNIT=2400,DISP=(OLD,KEEP),LABEL=(4,SL),DSN=WNTAIM,
// DCB=(RECFM=V,LRECL=3736,BLKSIZE=3740),VOL=SER=ERIC02
//MASTER DD UNIT=2400,DISP=(NEW,KEEP),LABEL=(1,SL)DSN=WNTAIM,
// DCB=(RECFM=V,LRECL=3736, BLKSIZE=3740),VOL=SER=ERIC04
```

SUBJECT QUARTERLY

1. Object Module

```
//FARM EXEC PGM=LOADER,PARM='LET',TIME=2
//SYSLIB DD DSN=SYS1.PL1LIB,DISP=SHR
//SYSLOUT DD SYSOUT=A
//SYSLIN DD *  quarterly subject object module
/*
//SYSPRINT DD SYSOUT=A
//ERRS DD SYSOUT=A
//MASTER DD UNIT=2400,DSN-WNTAIM,DISP=(OLD,KEEP),LABEL=(1,SL),
// DCB=(RECFM=V,LRECL=3736,FLKSIZE-3740),VOL=SER=ERIC04
//SORTLIB DD DSN=SYS1.SORTLIB,DISP=SHR
//SORTIN DD UNIT=SYSDA,DISP=(NEW,DELETE),SPACE=(CYL,(1,1)),
// DSN=UNSUB,DCB=(RECFM=V,LRECL=573,BLKSIZE=577)
//SORTOUT DD UNIT==SYSDA,DISP=(NEW,DELETE),SPACE=(CYL,(1,1)),
//DSN=STSUB,DCB=(RECFM=V,LRECL=573,BLKSIZE=577)
//SORTWK01 DD UNIT=SYSDA,SPACE=(TRK,20,,CONTIG)
//SORTWK02 DD UNIT=SYSDA,SPACE=(TRK,20,,CONTIG)
//SORTWK03 DD UNIT=SYSDA,SPACE=(TRK,20,,CONTIG)
//SORTWK04 DD UNIT=SYSDA,SPACE=(TRK,20,,CONTIG)
//SORTWK05 DD UNIT=SYSDA,SPACE=(TRK,20,,CONTIG)
//SORTWK06 DD UNIT=SYSDA,SPACE=(TRK,20,,CONTIG)
//SYSOUT DD SYSOUT=A
//OUTS DD SYSOUT=A,DCB=(RECFM=FA,LRECL=133,BLKSIZE=133)
//SUB DD UNIT=2400,DSN=AIMF2,DISP=(NEW,KEEP),LABEL=(1,SL),
//SYSIN DD * *** SYSIN DATA ON CARDS ***    control for page format
/*
```

OUTS is the index printout. When camera-ready is prepared, this card is replaced by DD card defining the tape data set.

2. Load Module

```
//JOBLIB DD DSN=ERICLIB,DISP=SHR
//SUBJ EXEC PGM=QSUBJ,TIME=4
//SYSPRINT DD SYSOUT=A
//ERRE DD SYSOUT=A
//MASTER DD UNIT=2400,DSN=WNTAIM,DISP=(OLD,KEEP),LABEL=(1,SL),
// DCB=(RECFM=V,LRECL=3736,BLKSIZE=3740),VOL=SER=ERIC04
//SORTLIB DD DSN=SYS1.SORTLIB,DISP=SHR
//SORTIN DD UNIT=SYSDA,DISP=(NEW,DELETE),SPACE=(CYL,(1,1)),
// DSN=UNSUB,DCB=(RECFM=V,LRECL=573,BLKSIZE=577)
//SORTOUT DD UNIT=SYSDA,DISP=(NEW,DELETE),SPACE=(CYL,(1,1),
//DSN=SISUB,DCB=(RECFM=V,LRECL=573,BLKSIZE=577)
//SORTWK01 DD UNIT=SYSDA,SPACE=(TRK,20,,CONTIG)
//SORTWK02      "
//SORTWK03      "
//SORTWK04      "
//SORTWK05      "
//SORTWK06      "
//SYSOUT DD SYSOUT=A
//OUTS DD SYSOUT=A,DCB=(RECFEM=FA,LRECL=133,BLKSIZE=133)
//SUB DD UNIT=2400,DSN=AIMF2,DISP=(NEW,KEEP),LABEL=(1,SL),
//SYSIN DD * *** SYSIN DATA ON CARDS ***      control for page format
/*
```

AUTHOR QUARTERLY

1. Object Module

```
//QAUTH EXEC PGM=LOADER,PARM='LET',TIME=2
//SYSLIB DD DSN=SYS1.PL1LIB,DISP=SHR
//SYSLOUT DD SYSOUT=A
//SYSLIN DD *    object deck for q. author
//SYSPRINT DD SYSOUT=A
//MASTER DD UNIT=2400,VOL=SER=ERIC02,DISP=OLD,DSN=SPRARM
//SORTIN DD UNIT=SYDA,SPACE=(CYL,(1,1)),DCB=(XECFM=V,LRECL=514,BLKSIZE=568)
//SORTOUT DD UNIT=2400,DSN=SPAUT,DCB=*.SORTIN,DISP=(KEEP),
//VOL=SER=ERIC20
//SORTWK01 DD UNIT=SYSDA,SPACE=(TRK,20,,CONTIG)
//SORTWK02 DD UNIT=SYSDA,SPACE=(TRK,20,,CONTIG)
//SORTWK03 DD UNIT=SYSDA,SPACE=(TRK,20,,CONTIG)
//SORTWK04 DD UNIT=SYSDA,SPACE=(TRK,20,,CONTIG)
//SORTWK05 DD UNIT=SYSDA,SPACE=(TRK,20,,CONTIG)
//SORTWK06 DD UNIT=SYSDA,SPACE=(TRK,20,,CONTIG)
//SYSOUT DD SYSOUT=A
//SORTLIB DD DSN=SYS1.SORTLIB,DISP=SHR
//OUT DD SYSOUT=A,DCB=(RECFM=VA,LRECL=133,BLKSIZE=137)
//SYSIN DD *    format control data
/*
```

OUT is index being created.

2. Load Module

```
//JOBLIB DD DSN=ERICLIB,DISP=SHR
//QAUTH EXEC PGM=QAUTHOR,TIME=2
//SYSPRINT DD SYSOUT=A
//MASTER DD UNIT=2400,VOL=SER=ERIC02,DISP=OLD,DSN=SPRARM
//SORTIN DD UNIT=SYSDA,SPACE=(CYL,(1,1)),DCB=(Xecfm=V,LRECL=514,BLKSIZE=568)
//SORTOUT DD UNIT=2400,DSN=SPAUT,DCB=*.SORTIN,DISP=(KEEP),
```

AUTHOR QUARTERLY (CONT'D.)

```
// VOL=SER=ERIC20
//SORTWK01 DD UNIT=SYSDA,SPACE=(TRK,20,,CONTIG)
//SORTWK02 DD UNIT=SYSDA,SPACE=(TRK,20,,CONTIG)
//SORTWK03 DD UNIT=SYSDA,SPACE=(TRK,20,,CONTIG)
//SORTWK04 DD UNIT=SYSDA,SPACE=(TRK,20,,CONTIG)
//SORTWK05 DD UNIT=SYSDA,SPACE=(TRK,20,,CONTIG)
//SORTWK06 DD UNIT=SYSDA,SPACE=(TRK,20,,CONTIG)
//SYSOUT DD SYSOUT=A
//SORTLIB DD DSN=SYS1.SORTLIB,DISP=SHR
//OUT DD SYSOUT=A,DCB=(RECFM=VA,LRECL=133,BLKSIZE=137)
//SYSIN DD *    format control data
/*
```

1. Object Module

```
//FARM EXEC PGM=LOADER,PARM='LET',TIME=2
//SYSLIB DD DSN=SYS1.PL1LIB,DISP=SHR
//SYSLOUT DD SYSOUT=A
//SYSLIN DD *    supporting services object module
/*
//SYSPRINT DD SYSOUT=A
//ERRE DD SYSOUT=A
//MASTER DD UNIT=2400,DSN=WNTAIM,LABEL=(4,SL),DISP=(OLD,KEEP),
// DCB=(RECFM=V,LRECL=3736,BLKSIZE3740),BOL=SER=ERIC02
//SORTLIB DD DSN=STS1.SORTLIB,DISP=SHR
//SORTIN DD UNIT=SYSDA,DISP=(NEW,DELETE),SPACE=(CYL,(1,1)),
// DSN=UNSUB,DCB=(RECFM=V,LRECL=444,BLKSIZE=448)
//SORTWKo1 DD UNIT=SYSDA,SPACE=(TRK,20,,CONTIG)
//SORTWKO2 DD UNIT=SYSDA,SPACE=(TRK,20,,CONTIG)
//SORTWKO3 DD UNIT=SYSDA,SPACE=(TRK,20,,CONTIG)
//SORTWKO4 DD UNIT=SYSDA,SPACE=(TRK,20,,CONTIG)
//SORTWKO5 DD UNIT=SYSDA,SPACE=(TRK,20,,CONTIG)
//SORTWKO6 DD UNIT=SYSDA,SPACE=(TRK,20,,CONTIG)
//SYSOUT DD SYSOUT=A
//OUTS DD SYSOUT=A,DCB=(RECFM=FA,LRECL=133,BLKSIZE=133)
//SUB DD UNIT=2400,DISP=(NEW,KEEP),LABEL=(1,SL),DSN=AIMF2,
// DCB=(RECFM=FB,LRECL=18,BLKSIZE=1800),VOL=SER=  RIC09
//SYSIN DD *    format control cards
/*
```

OUTS is the index being created

2. Load Module

```
//JOBLIB DD DSN=ERICLIB,DISP=SHR
```

SUPPORTING SERVICES QUARTERLY (cont'd.)

```
//QSS EXEC PGM=QSUPSER,TIME=2
//SYSPRINT DD SYSOUT=A
//ERRE DD SYSOUT=A
//MASTER DD UNIT=2400,DSN=WNTAIM,LABEL=(4,SL),DISP=(OLD,KEEP),
// DCB=(RECFM=V,LRECL=3736,BLKSIZE=3740),VOL=SER=ERIC 02
//SORTLIB DD DSN=STS1.SORTLIB,DISP=SHR
//SORTIN DD UNIT=SYSDA,DISP=(NEW,DELETE),SPACE=(CYL,(1,1)),
// DSN=UNSUB,DCB=(RECFM=V,LRECL=444,BLKSIZE=448)
//SORTWK01 DD UNIT=SYSDA,SPACE=(TRK,20,,CONTIG)
//SORTWK02 DD UNIT=SYSDA,SPACE=(TRK,20,,CONTIG)
//SORTWK03 DD UNIT=SYSDA,SPACE=(TRK,20,,CONTIG)
//SORTWK04 DD UNIT=SYSDA,SPACE=(TRK,20,,CONTIG)
//SORTWK05 DD UNIT=SYSDA,SPACE=(TRK,20,,CONTIG)
//SORTWK06 DD UNIT=SYSDA,SPACE=(TRK,20,,CONTIG)
//SYSOUT DD SYSOUT=A
//OUTS DD SYSOUT=A,DCB=(RECFM=FA,LRECL=133,BLKSIZE=133)
//SUB DD UNIT=2400,DISP=(NEW,KEEP),LABEL=(1,SL),DSN=AIMF2,
// DCB=(RECFM=FB,LRECL=18,BLKSIZE=1800),VOL=SER=ERIC09
//SYSIN DD *    format control cards
/*
```

DOCUMENT NUMBER QUARTERLY

1. Object Module

```
//FARM EXEC PGM=LOADER,PARM='LET',TIME=2
//SYSLIB DD DSN=SYS1.PL1LIB,DISP=SHR
//SYSLOUT DD SYSOUT=A
//SYSLIN DD *    document number object module
/*
//SYSPRINT DD SYSOUT=A
//SORTLIB DD DSNAME=SYS1.SORTLIB,DISP=SHR
//SORTIN DD DSNAME=INSORT,UNIT=SYSDA,DISP=(NEW,DELETE),
//SORTOUT DD DSNAME=OTSORT,UNIT=SYSDA,DISP=(NEW,DELETE),
//SORTWK01 DD UNIT=SYSDA,SPACE=(TRK,20,,CONTIG)
//SORTWK02 DD UNIT=SYSDA,SPACE=(TRK,20,,CONTIG)
//SORTWK03 DD UNIT=SYSDA,SPACE=(TRK,20,,CONTIG)
//SORTWK04 DD UNIT=SYSDA,SPACE=(TRK,20,,CONTIG)
//SORTWK05 DD UNIT=SYSDA,SPACE=(TRK,20,,CONTIG)
//SORTWK06 DD UNIT=SYSDA,SPACE=(TRK,20,,CONTIG)
//SYSOUT DD SYSOUT=A
//MASTER DD DSN=WNTAIM,UNIT=2400,VOL=SER=ERIC04,DISP=OLD,LABEL=(1,SL)
//ANNUAL DD DSN=ANNFIL,UNIT=2400,DISP=(NEW,KEEP),LABEL=(1,SL),
//          DCB=(RECFM=FB,LRECL=18,BLKSIZE=1800),VOL=SER=ERIC13
//INDIX DD SYSOUT=A,DCB=(RECFM=FBA,LRECL=133,BLKSIZE=133)
//SYSIN DD *    format control cards
/*
```

INDIX is the INDEX printout.
ANNUAL is annual subset file.

2. Load Module

```
//JOBLIB DD DSN=ERICLIB,DISP=SHR
//DOCNUM EXEC PGM=DOCNO,TIME=2
//SYSPRINT DD SYSOUT=A
```

DOCUMENT NUMBER QUARTERLY (cont'd.)

```
//SORTLIB DD DSNAME=SYS1.SORTLIB,DISP=SHR
//SORTIN DD DSNAME=INSORT,UNIT=SYSDA,DISP=(NEW,DELETE),
//SORTOUT DD DSNAME=OTSORT,UNIT=SYSDA,DISP=(NEW,DELETE),
//SORTWK01 DD UNIT=SYSDA,SPACE=(TRK,20,,CONTIG)
//SORTWK02 DD UNIT=SYSDA,SPACE=(TRK,20,,CONTIG)
//SORTWK03 DD UNIT=SYSDA,SPACE=(TRK,20,,CONTIG)
//SORTWK04 DD UNIT=SYSDA,SPACE=(TRK,20,,CONTIG)
//SORTWK05 DD UNIT=SYSDA,SPACE=(TRK,20,,CONTIG)
//SORTWK06 DD UNIT=SYSDA,SPACE=(TRK,20,,CONTIG)
//SYSOUT DD SYSOUT=A
//MASTER DD DSN=WNTAIM,UNIT=2400,VOL=SER=ERIC04,DISP=OLD,
//          LABEL=(1,SL)
//ANNUAL DD DSN=ANNFIL,UNIT=2400,DISP=(NEW,KEEP),LABEL=(1,SL),
//          DCB=(RECFM=FB,LRECL=18,BLKSIZE=1800),VOL=SER=ERIC13
//INDIX DD SYSOUT=A,DCB=(RECFM=FBA,LRECL=133,BLKSIZE=133)
//SYSIN DD *    format control cards
/*
```

CONVERSION OF DOCUMENT NUMBER

1. Object Deck

```
//FARM EXEC PGM=LOADER,PARM='LET',TIME=2
//SYSLIB DD DSN=SYS1.PL1LIB,DISP=SHR
//SYSLOUT DD SYSOUT=A
//SYSLIN DD *    conversion object deck
/*
//SYSPRINT DD SYSOUT=A
//SYSIN DD *    format control deck
/*
//OUTS DD SYSOUT=A,DCB=(RECFM=FA,LRECL=133,BLKSIZE=133)
//SORTLIB DD DSN=SYS1.SORTLIB,DISP=SHR
//SORTIN DD UNIT=2400,DISP=(OLD,KEEP),DSN=CONVRT,LABEL=(2,SLO,
// VOL=SER=ERIC04,DCB=(RECFM=FB,LRECL=20,BLKSIZE=200)
//SORTOUT DD UNIT=SYSDA,SPACE=(CYL,(1,1)),DISP=(NEW,DELETE),
// DSN=TEMPST,DCB=(RECFM=FB,LRECL=20,BLKSIZE=200)
//SORTWK01 DD UNIT=SYSDA,SPACE=(TRK,20,,CONTIG)
//SORTWK02 DD UNIT=SYSDA,SPACE=(TRK,20,,CONTIG)
//SORTWK03 DD UNIT=SYSDA,SPACE=(TRK,20,,CONTIG)
//SORTWK04 DD UNIT=SYSDA,SPACE=(TRK,20,,CONTIG)
//SORTWK05 DD UNIT=SYSDA,SPACE=(TRK,20,,CONTIG)
//SORTWK06 DD UNIT=SYSDA,SPACE=(TRK,20,,CONTIG)
//SYSOUT DD SYSOUT=A
```

OUTS is the index output.
SORTIN is accumulated VT-ED cards.
This program is used for both the quarterly and the annual conversion of document number index.

2. Load Module

```
//JOBLIB DD DSN=ERICLIB,DISP=SHR
```

CONVERSION OF DOCUMENT NUMBER (cont'd.)

```
//CONDOC EXEC PGM=CONVER,TIME=1
//SYSPRINT DD SYSOUT=A
//SYSIN DD *  format control deck
/*
//OUTS DD SYSOUT=A,DCB=(RECFM=FA,LRECL=133,BLKSIZE=133)
//SORTLIB DD DSN=SYS1.SORTLIB,DISP=SHR
//SORTIN DD UNIT=2400,DISP=(OLD,KEEP),DSN=CONVRT,LABEL=(s,SLO,
// VOL=SER=ERIC04,DCB=(RECFM=FB,LRECL=20,BLKSIZE=200)
//SORTOUT DD UNIT=SYSDA,SPACE=(CYL,(1,1)),DISP=(NEW,DELETE),
// DSN=TEMPST,DCB=(RECFM=FB,LRECL=20,BLKSIZE=200)
//SORTWK01 DD UNIT=SYSDA,SPACE=(TRK,20,,CONTIG)
//SORTWK02 DD UNIT=SYSDA,SPACE=(TRK,20,,CONTIG)
//SORTWK03 DD UNIT=SYSDA,SPACE=(TRK,20,,CONTIG)
//SORTWK04 DD UNIT=SYSDA,SPACE=(TRK,20,,CONTIG)
//SORTWK05 DD UNIT=SYSDA,SPACE=(TRK,20,,CONTIG)
//SORTWK06 DD UNIT=SYSDA,SPACE=(TRK,20,,CONTIG)
//SYSOUT DD SYSOUT=A
```

SUBJECT ANNUAL

1. Object Module

```
//FARM EXEC PGM=LOADER,PARM='LET',TIME=2
//SYSLIB DD DSN=SYS1.PLILIB,DISP=SHR
//SYSLOUT DD SYSOUT=A
//SYSLIN DD *     annual subject object deck
/*
//SYSPRINT DD SYSOUT=A
//SYSIN DD *      format control cards
/*
//SORTLIB DD DSN=SYS1.SORTLIB,DISP=SHR
//SORTIN DD UNIT=2400,DISP=(OLD,KEEP),LABEL=(1,SL),DSN=SB1,
//SORTOUT DD UNIT=SYSDA,DISP=(NEW,DELETE),SPACE=(CYL,(4,2)),
// DSN=SORTED,DCB=(RECFM=FB,LREC =140,BLKSIZE=1400)
//SORTWK01 DD UNIT=SYSDA,SPACE=(CYL(3))
//SORTWK02 DD UNIT=SYSDA,SPACE=(CYL(3))
//SORTWK03 DD UNIT=SYSDA,SPACE=(CYL(3))
//SORTWK04 DD UNIT=SYSDA,SPACE=(CYL(3))
//SORTWK05 DD UNIT=SYSDA,SPACE=(CYL(3))
//SORTWK06 DD UNIT=SYSDA,SPACE=(CYL(3))
//SYSOUT DD SYSOUT=A
//OUTS DD SYSOUT=A,DCB=(RECFM=FBA,LRECL=133,BLKSIZE=665)
```

OUTS is the data set for the index.

SUBJECT ANNUAL (cont'd.)

2. Load Module

```
//JOBLIB DD DSN=ERICLIB,DISP=SHR
//SUBJ EXEC PGM=ASUBJ,TIME=3
//SYSPRINT DD SYSOUT=A
//SYSIN DD *      format control cards
/*
//SORTLIB DD DSN=SYS1.SORTLIB,DISP=SHR
//SORTIN DD UNIT=2400,DISP=(OLD,KEEP),LABEL=(1,SL),DSN=SBL,
//SORTOUT DD UNIT=SYSDA,DISP=(NEW,DELETE),SPACE=(CYL,(4,2)),
// DSN=SORTED, DCB=(RECFM=FB,LRECL=140,BLKSIZE=1400)
//SORTWK01 DD UNIT=SYSDA,SPACE=(CYL,(3))
//SORTWK02 DD UNIT=SYSDA,SPACE=(CYL,(3))
//SORTWK03 DD UNIT=SYSDA,SPACE=(CYL,(3))
//SORTWK04 DD UNIT=SYSDA,SPACE=(CYL,(3))
//SORTWK05 DD UNIT=SYSDA,SPACE=(CYL,(3))
//SORTWK06 DD UNIT=SYSDA,SPACE=(CYL,(3))
//SYSOUT DD SYSOUT=A
//OUTS DD SYSOUT=A,DCB=(RECFM=FBA,LRECL=133,BLKSIZE=665)
```

AUTHOR ANNUAL

1. Object Module

```
//FARM EXEC PGM=LOADER,PARM='LET',TIME=2
//SYSLIB DD DSN=SYS1.PL1LIB,DISP=SHR
//SYSLOUT DD SYSOUT=A
//SYSLIN DD *
//SYSPRINT DD SYSOUT=A
//STEPLIB DD DSN=SYS1.JOBLIB,DISP=SHR
//SORTWK01 DD UNIT=SYSDA,SPACE=(TRK,20,,CONTIG)
//SORTWK02 DD UNIT=SYSDA,SPACE=(TRK,20,,CONTIG)
//SORTWK03 DD UNIT=SYSDA,SPACE=(TRK,20,,CONTIG)
//SORTWK04 DD UNIT=SYSDA,SPACE=(TRK,20,,CONTIG)
//SORTWK05 DD UNIT=SYSDA,SPACE=(TRK,20,,CONTIG)
//SORTWK06 DD UNIT=SYSDA,SPACE=(TRK,20,,CONTIG)
//SORTOUT DD UNIT=SYSDA, SPACE=(CYL,(1,1)),
//DCB=(RECFM=V,LRECL=564,BLKSIZE=568)
//SORTIN DD UNIT=2400,DCB=*.SORTOUT,VOL=(PRIVATE,RETAIN,SER=ERIC17),
// DISP=(OLD,PASS),DSN=ASRT
// DD UNIT=2400,VOL=SER=ERIC17,DCB=*.SORTOUT,DISP=(OLD,PASS),LABEL=2,
// DSN=AIMSPAUT
// DD UNIT=2400,VOL=SER=ERIC20,DSN=ARMSPAUT,DCB=*.SORTOUT,DISP=OLD
//SORTLIB DD DSN=SYS1.SORTLIB,DISP=SHR
//SYSOUT DD SYSOUT=A
//OUT DD SYSOUT=A
//SYSIN DD *
/*
```

2. Load Module

```
//JOBLIB DD DSN=ERICLIB,DISP=SHR
```

AUTHOR ANNUAL (cont'd.)

```
//AAUTH EXEC PGM=AAUTHOR,TIME=2
//SYSPRINT DD SYSOUT=A
//STEPLIB DD DSN=SYS1.JOBLIB,DISP=SHR
//SORTWK01 DD UNIT=SYSDA,SPACE=(TRK,20,,CONTIG)
//SORTWK02 DD UNIT=SYSDA,SPACE=(TRK,20,,CONTIG)
//SORTWK03 DD UNIT=SYSDA,SPACE=(TRK,20,,CONTIG)
//SORTWK04 DD UNIT=SYSDA,SPACE=(TRK,20,,CONTIG)
//SORTWK05 DD UNIT=SYSDA,SPACE=(TRK,20,,CONTIG)
//SORTWK06 DD UNIT=SYSDA,SPACE=(TRK,20,,CONTIG)
//SORTOUT DD UNIT=SYSDA,SPACE=(CYL,(1,1)),
// DCB=(RECFM=V,LRECL=564,BLKSIZE=568)
//SORTIN DD UNIT=2400,DCB=*.SORTOUT,VOL=(PRIVATE,RETAIN,SER=ERIC17),
// DISP=(OLD,PASS),DSN=ASRT
// DD UNIT=2400,VOL=SER=ERIC17,DCB=*.SORTOUT,DISP=(OLD,PASS).LABEL=2
// DSN=AIMSPAUT
// DD UNIT=2400,VOL=SER=ERIC20,DSN=ARMSPAUT,DCB=*.SORTOUT,DISP=OLD
//SORTLIB DD DSN=SYS1.SORTLIB,DISP=SHR
//SYSOUT DD SYSOUT=A
//OUT DD SYSOUT=A
//SYSIN DD *
/*
```

SUPPORTING SERVICES ANNUAL

1. Object Module

```
//FARM EXEC PGM=LOADER,PARM='LET',TIME=2
//SYSLIB DD DSN=SYS1.PL1LIB,DISP=SHR
//SYSLOUT DD SYSOUT=A
//SYSLIN DD *     annual s.s. object deck
/*
//SYSPRINT DD SYSOUT=A
//SYSIN DD *      format control cards
/*
//SORTLIB DD DSN==SYS1.SORTLIB,DISP=SHR
//SORTIN DD UNIT=2400,DISP=(OLD,KEEP),LABEL=(1,SL),DSN=TESTSB,
// DCB=(RECFM=FB,LRECL=18,BLKSIZE=1800),
//SORTOUT DD UNIT=SYSDA,DISP=(NEW,DELETE),SPACE=(CYL,(4,2)),
// DSN=SORTED,DCB=(RECFM=FB,LRECL=18,BLKSIZE=1800)
//SORTWK01 DD UNIT=SYSDA,SPACE=(CYL,(3))
//SORTWK02 DD UNIT=SYSDA,SPACE=(CYL,(3))
//SORTWK03 DD UNIT=SYSDA,SPACE=(CYL,(3))
//SORTWK04 DD UNIT=SYSDA,SPACE=(CYL,(3))
//SORTWK05 DD UNIT=SYSDA,SPACE=(CYL,(3))
//SORTWK06 DD UNIT=SYSDA,SPACE=(CYL,(3))
//SYSOUT DD SYSOUT=A
//OUTS DD SYSOUT=A,DCB=(RECFM=FA,LRECL=133,BLKSIZE=133)
```

2. Load Module

```
//JOBLIB DD DSN=ERICLIB,DISP=SHR
//ASSR EXEC PGM=ASYPSRV,TIME=3
//SYSPRINT DD SYSOUT=A
```

SUPPORTING SERVICES ANNUAL (cont'd.)

```
//SYSIN DD *      format control cards
/*
//SORTLIB DD DSN=SYS1.SORTLIB,DISP=SHR
//SORTIN DD UNIT=2400,DISP=(OLD,KEEP),LABEL=(1,SL),DSN=TESTSB,
// DCB=(RECFM=FB,LRECL=18,BLKSIZE=1800),
//SORTOUT DD UNIT=SYSDA,DISP=(NEW,DELETE),SPACE=(CYL,(4,2)),
// DSN=SORTED,DCB=(RECFM=FB,LRECL=18,BLKSIZE=1800)
//SORTWK01 DD UNIT=SYSDA,SPACE=(CYL,(3)),
//SORTWK02 DD UNIT=SYSDA,SPACE=(CYL,(3)),
//SORTWK03 DD UNIT=SYSDA,SPACE=(CYL,(3)),
//SORTWK04 DD UNIT=SYSDA,SPACE=(CYL,(3)),
//SORTWK05 DD UNIT=SYSDA,SPACE=(CYL,(3)),
//SORTWK06 DD UNIT=SYSDA,SPACE=(CYL,(3)),
//SYSOUT DD SYSOUT=A
//OUTS DD SYSOUT=A,DCB=(RECFM=FA,LRECL=133,BLKSIZE=133)
```

DOCUMENT NUMBER ANNUAL

1. Object Module

```
//VTISSUE EXEC PGM=LOADER,PARM='LET',TIME=2
//SYSLIB DD DSN=SYS1.PL1LIB,DISP=SHR
//SYSLOUT DD SYSOUT=A
//SYSIN DD *      VT-issue object deck
/*
//SYSPRINT DD SYSOUT=A
//SYSIN DD *      control parameters
/*
//SORTLIB DD DSN=SYS1.SORTLIB,DISP=SHR
//SORTIN DD UNIT=2400,DISP=OLD,DSN=ANSUB1,
// DCB=(RECFM=FB,LRECL=18,BLKSIZE=1800),VOL=SER=ERIC
//SORTOUT DD UNIT=SYSDA,SPACE=(CYL,(1,1)),DCB=*.SORTIN
//SORTWK01 DD UNIT=SYSDA,SPACE=(TRK,20,,CONTIG)
//SORTWK02 DD UNIT=SYSDA,SPACE=(TRK,20,,CONTIG)
//SORTWK03 DD UNIT=SYSDA,SPACE=(TRK,20,,CONTIG)
//SORTWK04 DD UNIT=SYSDA,SPACE=(TRK,20,,CONTIG)
//SORTWK05 DD UNIT=SYSDA,SPACE=(TRK,29,,CONTIG)
//SORTWK06 DD UNIT=SYSDA,SPACE=(TRK,20,,CONTIG)
//SYSOUT DD SYSOUT=A
//OUTS DD SYSOUT=A
```

2. Load Module

```
//JOBLIB DD DSN=ERICLIB,DISP=SHR
//VTISSUE EXEC PGM=ADNO,TIME=1
//SYSPRINT DD SYSOUT=A
//SYSIN DD *      control parameters
/*
//SORTLIB DD DSN=SYS1.SORTLIB,DISP=SHR
//SORTIN DD UNIT=2400,DISP=OLD,DSN=ANSUB1,
```

DOCUMENT NUMBER ANNUAL (cont'd.)

```
// DCB=(RECFM=FB,LRECL=18,BLKSIZE=1800),VOL=SER=ERIC
//SORTOUT DD UNIT=SYSDA,SPACE=(CYL,(1,1)),DCB=*.SORTIN
//SORTWK01 DD UNIT SYSDA,SPACE=(TRK,20,,CONTIG)
//SORTWK02 DD UNIT SYSDA,SPACE=(TRK,20,,CONTIG)
//SORTWK03 DD UNIT SYSDA,SPACE=(TRK,20,,CONTIG)
//SORTWK04 DD UNIT SYSDA,SPACE=(TRK,20,,CONTIG)
//SORTWK05 DD UNIT SYSDA,SPACE=(TRK,20,,CONTIG)
//SORTWK06 DD UNIT SYSDA,SPACE=(TRK,20,,CONTIG)
//SYSOUT DD SYSOUT=A
//OUTS DD SYSOUT=A
```

EDIT OF VOCATIONAL, AND SUPPORTING SERVICES, SUBJECT, AND AUTHOR

1. Object Module

```
//FARM EXEC PGM=LOADER,PARM='LET',TIME=2
//SYSLIB DD DSN=SYS1.PL1LIB,DISP=SHR
//SYSLOUT DD SYSOUT=A
//SYSLIN DD *     edit object deck
/*
//FIX DD UNIT=2400,DISP=(OLD,KEEP),LABEL=(1,SL),DSN=QRTVC1,
// VOL=(PRIVATE,RETAIN,SER=ERIC03)
//TEMP DD UNIT=2400,DISP (NEW,KEEP),LABEL=(1,SL),DSN=QRTVC1,
// DCB=(RECFM=FB,LRECL=18,BLKSIZE=1800),VOL=(,RETAIN,SER=ERIC20)
```

FIX is the data set to be edited.

TEMP is the edited data set.

2. Load Module

```
//JOBLIB DD DSN=ERICLIB,DISP=SHR
//EDIT EXEC PGM=SUBJSSA,TIME=2
//SYSPRINT DD SYSOUT=A
//SYSIN DD *   control and edit data
/*
//FIX DD UNIT=2400,DISP=(OLD,KEEP),LABEL=(1,SL),DSN=QRTVC1,
// VOL=(PRIVATE,RETAIN,SER=ERIC03)
//TEMP DD UNIT=2400,DISP=(NEW,KEEP),LABEL=(1,SL)DSN=QRTVC1,
// DCB=(RECFM=FB,LRECL=18,BLKSIZE=1800),VOL=(,RETAIN,SER=ERIC20)
```

DOCUMENT NUMBER EDIT

1. OBJECT MODULE

```
// EXEC PGM=LOADER,PARM='LET;TIME=2
//SYSLIB DD DSN=SYS1.PL1LIB,DISP=SHR
//SYSLOUT DD SYSOUT=A
//SYSLIN DD *     edit object deck
/*
//DISC DD UNIT=SYSDA,DSN=&&EDTOUT,DISP=(OLD,PASS),
// DCB=(RECFM=FB,LRECL=18,BLKSIZE=1800)
//TAPE DD UNIT=2400,VOL=SER=ERIC30,DISP=(NEW,PASS),
// DSN=EDITAPE,DCB=(RECFM=FB,LRECL=18,BLKSIZE=1800)
//CARDS DD *     data
/*
```

DISC is the data set to be edited.

TAPE is the edited data set.

2. LOAD MODULE.

```
//JOBLIB DD DSN=ERICLIB,DISP=SHR
//EDIT EXEC PGM=VTISSUE,TIME=2
//SYSPRINT DD SYSOUT=A
//CARDS DD *     edit data
/*
//DISC DD UNIT=SYSDA,DSN=&&EDTOUT,DISP=(OLD,PASS),
// DCB=(RECFM=FB,LRECL=18,BLKSIZE=1800)
//TAPE DD UNIT=2400,VOL=SER=ERIC30,DISP=(NEW,PASS),
// DSN=EDITAPE,DCB=(RECFM=FB,LRECL=18,BLKSIZE=1800)
```

RIE FORMAT

1. Object Module

```
//FARM EXEC PGM=LOADER,PARM='LET',TIME=2
//SYSLIB DD DSN=SYS1.PL1LIB,DISP=SHR
//SYSLIN DD *
/*
//SYSPRINT DD SYSOUT=A
//MASTER DD UNIT=2400,VOL=SER=ERIC26,DISP=(MOD,KEEP),DSN=ARIM6970,
// DCB=(RECFM=V,LRECL=6996,BLKSIZE=7000)
//RIE DD UNIT=2400,VOL=SER=ERIC04,DSN=WNTAIM,
// DCB=(RECFM=V,LRECL=3736,BLKSIZE=3740),DISP=(OLD,PASS)
//ABSTRACT DD UNIT=2400,VOL=SER=ERIC,DISP=OLD,
// DCB=(RECFM=V,LRECL=2000,BLKSIZE=2004)
```

MASTER is the final data set. The format will be the same as RIE and this will be the search data set.

RIE is the data set containing bibliographic data.

ABSTRACT is the abstract records for the appropriate records.

2. Load Module

```
//JOBLIB DD DSN=ERICLIB,DISP=SHR
//RIEFORM EXEC PGM=VTTORIE,TIME=1
//SYSPRINT DD SYSOUT=A
//MASTER DD UNIT=2400,VOL=SER=ERIC26,DISP=(MOD,KEEP),DSN=ARIM6970
// DCB=(RECFM=V,LRECL=6996,BLKSIZE=7000)
//RIE DD UNIT=2400,VOL=SER=ERIC04,DSN=WNTAIM,
// DCB=(RECFM=V,LRECL=3736,BLKSIZE=3740),DISP=OLD
//ABSTRACT DD UNIT=2400,VOL=SER=ERIC,DISP=OLD
// DCB=(RECFM=V,LRECL=2000,BLKSIZE=2004)
```

APPENDIX C

CONTROL DATA CARDS

	page
RIE TO AIM/ARM	37
MASTER	38
ABSTRACT	39
RESUME PRINT	40
MASTER EDIT	41
SUBJECT QUARTERLY	42
AUTHOR QUARTERLY	43
SUPPORTING SERVICES QUARTERLY	44
DOCUMENT NUMBER QUARTERLY	45
CONVERSION OF DOCUMENT NUMBER	46
SUBJECT ANNUAL	47
AUTHOR ANNUAL	48
SUPPORTING SERVICES ANNUAL	49
DOCUMENT NUMBER ANNUAL	50
EDIT OF SUBJECT, AUTHOR, & SUPPORTING SERVICES	51
DOCUMENT NUMBER EDIT	52

RIE TO AIM/ARM

There are two card data sets for this program.

1. SYSIN

The data is input in a free format. The first character is a blank on an 'A'. The blank will use columns 23 and 24 of the code card for the abstract code. A letter 'A' will use columns 20 and 21 for the abstract code.

Next is the numeric giving the maximum number of documents to be extracted from the tape. The third item is the 1 to 7 character string used to denot the issue to be published.

2. VTSIN

This data set consists of an accession number in columns 1 through 10 in the form VT 999 999, and the form, field and category in columns 15 through 32. This code will be in the bibliographic record and the two characters designated by the SYSIN card will be in the abstract record.

MASTER

A. CARD 1: all fields are F(3,0)

These are highest line values for the various sections:

Variable		Section	Current Value (max. names)
1.	NA	Number of Personal Authors	2
2.	N10	Personal author	10
3.	NT	Title	22
4.	NC	Corporate author	34
5.	NS	Sponsoring agency	42
6.	N50	Price section	50
7.	N60	Report and bureau	60
8.	N70	Availability	71
9.	N80	Journal citation	80
10.	N90	Date, page, desc. note	91
11.	N100	Descriptor	104
12.	N200	Identifier	202

B. CARD 2: NUMBER OF DOCUMENTS TO BE PROCESSED (USED AS A LIMIT NOT ACTUAL COUNT) right justify in col. 1-5

C. CARD 3:

1. col. 1, A for AIM, blank for ARM
2. col. 2-3, SEASON 'F ', 'W ', 'SP', or 'SU'
3. col. 4-7, current year 19__.

D. CARDS 4-?

1. col. 1-10, VT number (i.e. Clearinghouse accession number)
2. col. 11-30, ED or accession number
3. col. 31-34, resume page

E. Blank card

ABSTRACT TAPE CREATION

Data consists of one card with an A in column 1 if the abstracts should contain the sixth and seventh characters of the code. Column 1 contains a blank if the ninth and tenth characters are to be used.

RESUME SECTION PRINT PROGRAM

The data set is contained in three parts:

Part 1: one card containing A_OR_BLNK='A'; or A_OR_BLNK='ϕ' ('A' for AIM, 'ϕ' for ARM)

Part 2: several cards in the form variable=value used for print control

Variable	Usage	Current Value
PRICE_COL	Print column for price data	9
CONTRACT_COL	Space from price to CONTRACT number	5
GRANT_COL	Space from contract to grant number	5
PA_COL	Space from contract or grant number to program area	5
VT_ED_SKIP	Space between VT and ED number	5
TOTAL_COLUMNS	Maximum print positions	124
TOTAL_LINES	Length of page	77
PAGE-WIDTH	Actual count of print characters	88
TOP	Indicate right or left page	'RITE'
VT_COL	Print column for VT no.	9
FIRST_LINE	First line of resume section print on first section page	1
CON_1ST_LINE	First line of resume section print on succeeding pages	1
INDENT	Columsn to be left for margin	8
SPACE_LINES	Number of lines to be skipped between title and corporate author, between the descr. note and descriptors, and between the identifiers and abstract.	1
HEADING	Publication	'ARM/FALL 70'
INDEX_PAGE	First page of resume section	2

Part 3: documents to be deleted.

This data is in free format. The first value is number of documents to be deleted. The maximum number is 20. The remainder of this data consist of the clearinghouse accession numbers contained in quotes.

MASTER EDIT

A. Card 1: number of documents (N) to be edited, right justified in columns 1-4.

B. Cards 2-N+1: the clearinghouse accession numbers of documents to be edited in columns 1-10.

C. Card N+2: all fields are F(3,0).

These are the highest line values for the various sections or maximum number of names:

Variable	Section	Current Value
NA	Personal author	2
N10	"	10
NT	Title	22
NC	Corporate author	34
NS	Sponsoring agency	42
N50	Price section	50
N60	Report and Bureau	60
N70	Availability	71
N80	Journal citation	80
N90	Date, page, desc note	91
N100	Descriptors	104
N200	Identifiers	202

D. Card N+3: Total number of documents expected to be processed. This number is an upper limit and the current value is 600.

E. Card N+4: Card to indicate the issue:

1. Column 1 contains an A if AIM and a blank if B
2. Columns 2-3 -- F for FALL, W for WINTER, SP for SPRING, or SU for SUMMER.
3. Columns 4-7 -- the current year 19??

F. Cards N+5 to end of deck: clearinghouse accession number, ED number, and page number of the resume.

SUBJECT QUARTERLY

Data is one continuous stream and varies in length.

Field	Format	Contents	Current Value
1	F(4,0)	First page of index	552
2	F(4,0)	Size of page (total lines)	77
3	F(4,0)	Column width	37
4	F(4,0)	Left hand page margin	9
5	F(4,0)	Right hand page margin	99
6	F(4,0)	Title page line for page number and heading	2
7	F(4,0)	Title page line for word "page"	9
8	F(4,0)	Line of succeeding pages for page number and heading	2
9	F(4,0)	Line of succeeding pages for title	4
10	F(4,0)	Line of succeeding pages for word "page"	7
11	F(4,0)	Spaces between VT and ED numbers	5
12	F(4,0)	Number of columsn in index printout	2
13	F(4,0) repeated no. of cols. times	Margin for each column measured from column one	9,54
14	F(4,0)	Length of title	28
15	A(field 14's value)	Index title	SUBJECT AND IDENTIFIER INDEX
16	F(4,0)	Length of heading	13
17	A(field 16's value)	Heading (e.g. Fall/ARM 70)	WINTER/ARM 69
18	A(1)	Which side is first page on (L for left, R for right)	L

AUTHOR QUARTERLY

Data is in format: variable name = value.

Variable Name	Variable Usage	Current Values
1. #_COLS	Number of columns	2
2. MARGIN	Total print space for authors' name	37
3. LEN_PAGE	Number of print lines	77
4. HEADING	(e.g. AIM/FALL 69)	AIM/WINTER 69
5. TOP	Which side is the page (e.g. TOP='LEFT' the index page number is on left)	LEFT
6. COL SPACE	Character string on blanks to be placed between columns	'/bbbb'
7. VT_ED_SKIP	Number of spaces to be skipped between VT and ED numbers	5
8. INDEX PAGE	Initial page number	892
9. INITIAL LINE	Line for 'PAGE' on title page	11
10. NEXT INITIAL	Line for 'PAGE' on succeeding pages	9

SUPPORTING SERVICES QUARTERLY

Data is one continuous stream and varies in length.

Field	Format	Contents	Current Values
1	F(4,0)	Initial page	917
2	F(4,0)	Length of pages	77
3	F(4,0)	Column width index printout	37
4	F(4,0)	Left hand margin	9
5	F(4,0)	Right hand margin	93
6	F(4,0)	Title page line number for page number and heading	2
7	F(4,0)	Title page line number for 'PAGE'	9
8	F(4,0)	Succeeding pages line number for page number and heading	2
9	F(4,0)	Succeeding pages line number for title	4
10	F(4,0)	Succeeding pages line number for 'PAGE'	7
11	F(4,0)	Number of spaces between VT and ED numbers	5
12	F(4,0)	Number of columns per page	2
13	F(4,0) repeated no. of cols. times	Column margin from column one	9,54
14	F(4,0)	Length of title	40
15	A(value of field 14)	Index title	VOCATIONAL AND SUPPORTING SERVICES INDEX
16	F(4,0)	Length of heading	13
17	A(value of field 16)	Heading	ARM/WINTER 69
18	A(1)	Indicates whether page number is in left (L) or right (R) margin	R

DOCUMENT NUMBER QUARTERLY

There are two print control cards necessary.

A. CARD 1

Columns	Contents	Current Values
1-3	AIM or ARM	AIM
4-6	line number of first line INDX, SEASON, Year, PAGE	2
7-10	first page number	927
11-13	length of page (i.e. print lines)	77
14-16	first page left or right (l=left, r=right)	2
17-19	width of margin 1	8
20-22	width of margin 2 (0 if no margin)	5
23-25	width of margin 3 (0 if no margin)	0
26-28	width of column (from VT in VT to E in PAGE)	40
29-31	number of columns	2
[illegible]	number of spaces between VT number and ED number	5
35-37	line number of line containing "PAGE"	7
38-40	line number of first VT ED line	9
41-43	line number of title page for "PAGE"	9
44-46	line number of title page for VT ED	11

B. CARD 2

Columns	Contents
1-3	length of title
4-6	line number for title
7-80	title

CONVERSION OF DOCUMENT NUMBER

Data is one continuous stream and varies in length. This program is used for both annual and quarterly index.

Field	Format	Contents	Current Values
1	F(4,0)	initial page	600
2	F(4,0)	number of print lines	93
3	F(4,0)	column width	25
4	F(4,0)	left hand margin	9
5	F(4,0)	right hand margin	102
6	F(4,0)	line for page number on title page	2
7	F(4,0)	line for subtitle on title page	9
8	F(4,0)	line on succeeding pages for page number	2
9	F(4,0)	line on succeeding pages for title	4
10	F(4,0)	line on succeeding pages for subtitle	7
11	F(4,0)	lines skipped at bottom of page before footnote	3
12	F(4,0)	number of columns	3
13	F(4,0) repeated no. of cols. times	column margins measured from column one	3,9
14	F(4,0)	number of lines of print per block	42
15	F(4,0)	number of blocks of print per columns on title page	75
16	F(4,0)	number of blocks of print per column on succeeding pages	5
17	F(4,0)	length of title	35
18	A(value of field 17)	index title	CONVERSION OF DOCUMENT NUMBER INDEX
19	F(4,0)	length of heading	13
20	A(value of field 19)	heading stating issue, reason, and year	AIM/ARM 69-70
21	A(1)	Q for quarterly index, A for annual index	A
22	A(1)	L if title page is on left, R if title page is on right	L

SUBJECT ANNUAL

Data is one continuous stream and varies in length.

Field	Format	Contents	Current Value
1	F(4,0)	First page of index	1
2	F(4,0)	Size of page (total lines)	95
3	F(4,0)	Column width	34
4	F(4,0)	Number significant digits for VT number	5
5	F(4,0)	Left hand page margin	9
6	F(4,0)	Right hand page margin	125
7	F(4,0)	Title page line for page number and heading	2
8	F(4,0)	Line on title page directly above first line of index data	10
9	F(4,0)	Line of succeeding pages for page number and heading	2
10	F(4,0)	Line of succeeding pages for title	4
11	F(4,0)	Line on succeeding pages directly above first line of index.	7
12	F(4,0)	Number of columns in index printout	3
13	F(4,0) repeated no. of cols. times	Margin for each column measured from column one	9
14	F(4,0)	Length of title	49
15	A(field 14's value)	Index title	SUBJECT AND IDENTIFIER INDEX
16	A(13)	Heading "AIM/ARM 69-70"	
17	A(1)	L if title page is on left, R if title page is on right	R

AUTHOR ANNUAL

Data is in format: variable name=value.

Variable Name	Variable Usage	Current Value
MARGIN	Width of print	37
VT_WIDTH	Columns allocated to VT no.	5
DATE	Last 2 digits of present year	70
INDEX_PAGE	Initial page number	280
#_COLS	Number of columns	2
PRINT_LINE	Number of print positions across page	95
PAGE_LINES	Length of page	77
HEADING	E.g., "AIM/ARM 69-70"	"AIM/ARM 69-70"
TOP	LEFT if initial page is on left RITE if initial page is on right	"RITE"
FIRST_INITIAL	First line of index on title page	9
NEXT_INITIAL	First line of index on succeeding pages	7
COL_SPACE	String of blanks to be placed between columns	'␣␣␣␣␣'

SUPPORTING SERVICES ANNUAL

Data is one continuous stream and varies in length.

Field	Format	Contents	Current Value
1	F(4,0)	First page of index	93
2	F(4,0)	Size of page (total lines)	97
3	F(4,0)	Column width	34
4	F(4,0)	Number significant digits for VT number	5
5	F(4,0)	Left hand page margin	9
6	F(4,0)	Right hand page margin	125
7	F(4,0)	Title page line for page number and heading	2
8	F(4,0)	Line on title page directly above first line of index data	10
9	F(4,0)	Line of succeeding pages for page number and heading	2
10	F(4,0)	Line of succeeding pages for title	4
11	F(4,0)	Line on succeeding pages directly above first line of index	7
12	F(4,0)	Number of columsn in index printout	3
13	F(4,0) repeated no. of cols. times	Margin for each column measured from column one	9
14	F(4,0)	Lenth of title	49
15	A(field 14's value)	Index title	VOCATIONAL AND SUPPORTING SERVICES INDEX
16	A(13)	Heading "AIM/ARM 69-70"	
17	A(1)	L if title page is on left, R if title page is one right	R

DOCUMENT NUMBER ANNUAL

Field	Format	Contents	Current Value
1	F(4)	Number of index page	97
2	F(4)	Maximum length of page	97
3	F(4)	No. of significant digits in VT no.	5
4	F(4)	Left hand margin	10
5	F(4)	Right hand margin	124
6	F(4)	Line for heading and page number	2
7	F(4)	Line for subtitle	11
8	F(4)	Line for heading and page number on succeeding pages	2
9	F(4)	Line for index title on all but title page	4
10	F(4)	Line for subtitle on succeeding pages	8
11	F(4)	Number of columns between VT number and issue in printout	5
12	F(4)	Number of columns between issue and page number in printout	5
13	F(4)	Number of columns in index printout	3
14	Field 13's Value*F(4)	Array of column margins measured from column one.	15, 55, 95
15	F(4)	Number of lines of output in one block	5
16	F(4)	Number of blocks of output on title page	14
17	F(4)	Number of blocks of output on other pages	14
18	F(4)	Length of index title	39
19	A (Field 18's value)	Title of index	*
20	A(13)	Heading	AIM/ARM 67-70
21	A(1)	Whether title page shall be on left or right	R

* VT number, issue, and page number index.

EDIT OF SUBJECT, AUTHOR, AND SUPPORTING SERVICES

Card 1: free format first field a numeric indicating the length of the keyword. (e.g. Author edit the value is 132). This field is followed by a blank and then 4 characters in quotes to indicate the index to be edited. If the index is Subject, use "SUBJ."

Card 2: number of records to be replaced followed by the number of records to be deleted.

Card 3:to end of replacements: keyword, date and VT number of error followed by keyword, date, and VT number of correct record. One set for each document.

Cards following replacement deck: deletion deck containing keyword, date, and VT number for documents to be deleted.

Cards following deletion deck: first a card containing number of documents to be added. This card is followed by a deck of cards containing keyword, date, and VT number for each document to be added.

DOCUMENT NUMBER EDIT

Format of input to edtindex

VT number	Columns 1-6
Page No. (right justified)	Columns 7-10
Issue	Columns 11-15
Card Code	Column 80

If card code is a blank, all corresponding entries on the input data set will be deleted when written to output data set.

If card code is an A, columns 1-18 of that input card will be used as the entry to be written to the output data set.

Sort Input Cards

Numeric Sort	Columns 1-6

APPENDIX D

ERROR MESSAGES FROM PROGRAMS

	page
MASTER	54
ABSTRACT	54
RESUME PRINT	54
MASTER EDIT	55
SUBJECT QUARTERLY	55
AUTHOR QUARTERLY	56
SUPPORTING SERVICES QUARTERLY	56
SUBJECT ANNUAL	56
AUTHOR ANNUAL	56
SUPPORTING SERVICES ANNUAL	57

MASTER PROGRAM MESSAGES

1. CODE NO, IS OUT OF BOUNDS VT ___ ___. Columns 6 and 7 of code are not between 40 and 52. The limits are set in card 3440 of the program.

2. SHORT TITLE MARKS ARE NOT CORRECT IN DOCUMENT VT ___ ___.

3. MISMATCHED CARD NUMBER IN DOCUMENT VT ___ ___.

4. MISSING END OF LINE OR TAB CHARACTER IN VT ___ ___.

5. LENGTH OF VARIABLE OR DIMENSION IS OFF IN VT ___ ___. This message is used for character strings which are too long for present sizes or there are too many entries, e.g. more than 5 corporate authors would cause this under current limits.

6. DOCUMENT NO. DOES NOT MATCH ON ONE CARD IN VT ___ ___.

7. EXPECTED INFORMATION MISSING FROM DOCUMENT VT ___ ___. This may be caused by some cards being out of order.

8. NO MATCH MADE ON ED NUMBER FOR VT ___ ___. The clearinghouse accession number card for this document is missing.

ABSTRACT

Cards from abstract record indicate there was a mistake in the abstract card deck.

RESUME PRINT

1. ERROR ON MATCH VT ___ ___ VT ___ ___. Trouble in matching bibliographic record with abstract record.

2. ERRORS IN VTS RESUME BIB VT ___ ___ VT ___ ___. Trouble in matching abstract record with bibliographic record.

3. TITLE ERROR bibliographic record. Title marks are not correct in the bibliographic record printed.

MASTER EDIT

1. CODE NO. IS OUT OF BOUNDS VT ___ ___. Columns 6 and 7 of code are not between 40 and 52. The limits are set in card 3440 of the program.

2. SHORT TITLE MARKS ARE NOT CORRECT IN DOCUMENT VT ___ ___.

3. MISMATCHED CARD NUMBER IN DOCUMENT VT ___ ___.

4. MISSING END OF LINE OR TAB CHARACTER IN VT ___ ___.

5. LENGTH OF VARIABLE OR DIMENSION IS OFF IN VT ___ ___. This message is used for character strings which are too long for present sizes or there are too many entries, e.g. more than 5 corporate authors would cause this under current limits.

6. DOCUMENT NO. DOES NOT MATCH ON ONE CARD IN VT ___ ___.

7. EXPECTED INFORMATION MISSING FROM DOCUMENT VT ___ ___. This may be caused by some cards being out of order.

8. NO MATCH MADE ON ED NUMBER FOR VT ___ ___. The clearinghouse accession number card for this document is missing.

SUBJECT QUARTERLY

1. MISSING DIVIDER IN DOCUMENT VT ___ ___. The divider separating the numeric length fields from the character string is missing in the bibliographic record.

2. KEYWORD TOO LARGE IN VT ___ ___. The present limit on the descriptor and identifier length is set at 200 characters and a keyword in the designated document is too long.

3. TROUBLE IN BREAK WITH VT ___ ___. This error message indicates that there is no blank or - to break up the descriptor or identifier for this document.

AUTHOR QUARTERLY

1. INDEX IS WRONG index number. Separator in bibliographic record is missing.
2. PERSONAL AUTHOR WRONG numeric. Number of personal authors exceeds two.
3. CORPORATE AUTHOR WRONG numeric. Number of corporate authors exceeds five.
4. SPONSORING WRONG numeric. Number of sponsoring agencies exceeds three.
5. TITLE ERROR title VT ___ ___. Error in the title marks.

SUPPORTING SERVICES QUARTERLY

MISSING DIVIDER IN DOCUMENT VT ___ ___. The divider seperating the numeric length fields from the character string is missing in the bibliographic record.

SUBJECT ANNUAL

1. NO END OF SUBJECT NAME IN subject VT ___ ___. Special character used for end of subject name is missing from the record.
2. MORE THAN 15 DATES UNDER service field. Present limit on number of dates for a service field is 15 dates.
3. MORE THAN 75 DOCUMENTS UNDER publication date FOR KEYWORD ____.

AUTHOR ANNUAL

Bibliographic record printed out indicates error in record.

SUPPORTING SERVICES ANNUAL

1. MORE THAN 15 DATES UNDER service field. Present limit on number of dates for a service field is 15 dates.

2. MORE THAN 200 DOCUMENTS UNDER publication date FOR KEYWORD _____.

APPENDIX E

PROGRAM LISTINGS

	page
RIE TO AIM/ARM	59
MASTER	69
ABSTRACT	84
RESUME PRINT	87
MASTER EDIT	95
SUBJECT QUARTERLY	106
AUTHOR QUARTERLY	120
SUPPORTING SERVICES QUARTERLY	128
DOCUMENT NUMBER QUARTERLY	141
CONVERSION OF DOCUMENT NUMBER	150
SUBJECT ANNUAL	159
AUTHOR ANNUAL	171
SUPPORTING SERVICES ANNUAL	178
DOCUMENT NUMBER ANNUAL	190
EDIT OF SUBJECT, AUTHOR, AND SUPPORTING SERVICES	197
DOCUMENT NUMBER EDIT	200
RIE FORMAT	202

RIE TO AIM/ARM

```
/*
                G E N E R   P R O G R A M

      PURPOSE:  REFORMAT R I E TAPES INTO FORMATS FOR BIBLIOGRAPHIC
                AND ABSTRACT TAPES

                M A S T E R  R E C O R D  F O R M A T
*/

      1.  VT NUMBER          10 BYTES
      2.  CODE NUMBER        17 BYTES
      3.  ED NUMBER          20 BYTES
      4.  PUB. ISSUE          3 BYTES       A OR BLANK AND 2 BYTH QUARTER
      5.  CURRENT YEAR        4 BYTES
      6.  RESUME PAGE         4 BYTES
      7.  PERSONAL AUTHOR   132 BYTHES MAXIMUM   (MAY BE 2 AUTHORS)
      8.  TITLE             396 BYTES MAXIMUM
      9.  CORP. AUTHORS     660 BYTES MAXIMUM    (MAY BE 5 AUTHORS)
     10.  SPONSOR. AGENCY   396 BYTES MAXIMUM    (MAY BE 3 AGENCIES)
     11.  PRICE              66 BYTES MAXIMUM
     12.  CONTRACT NO.       44 BYTES MAXIMUM
     13.  GRANT NO.          44 BYTES MAXIMUM
     14.  PA NO.              2 BYTES IF IT EXISTS
     15.  REPORT NO.         66 BYTES MAXIMUM
     16.  BUREAU NO.         66 BYTES MAXIMUM
     17.  AVAILABILITY      264 BYTES MAXIMUM
     18.  JOURNAL CITAT.    132 BYTES MAXIMUM
     19.  PUBLICATION DATE    7 BYTES          PUBLICATION OF THE DOCUMENT
     20.  NO. OF DOC. PP.    16 BYTES MAXIMUM
     21.  DESCRIPTOR NOTE   247 BYTES MAXIMUM
     22.  DESCRIPTORS       660 BYTES MAXIMUM
     23.  IDENTIFIERS       396 BYTES MAXIMUM
     24.  SEPARATOR BETWEEN CHARACTER FIELDS AND LENGTH INDICATORS 2 VERTI
          EACH VALUE OCCUPIES 4 BYTES UNLESS NOTED THE FIELD CONTAINS THE LE
          THE INDICATED FIELD
     25. NUMBER OF PERSONAL AUTHORS
     27. TITLE
     28. CORPORATE AUTHOR (NUMBER OF)
     29. CORPORATE AUTHORS  0 TO 5 FIELDS
     30. SPONSORING AGENCIES (NUMBER OF)
     31. SPONSORING AGENCIES 0 TO 3 FIELDS
     32. PRICE
     33. CONTRACT
     34. GRANT NO.
     35. PA NO.
     36. REPORT NO.
     37. BUREAU NO.
     38. AVAILABILITY
     26. EACH P. AUTHOR 0 TO 2 FIELDS
     39.  JOURNAL CITATION
     40.  PUBLICATION DATE
     41.  PAGE COUNT
     42.  DESCRIPTIVE NOTE
     43.  DESCRIPTORS
     44.  IDENTIFIERS

*/
/*
          C O D E          T A B L E
     SECTION               HEX. CODE      DECIMAL CODE      ORDER
ACCESSION NUMBER             0010             16              1
```

```
   CLEARINGHOUSE NUM.          0011          17          2    00000630
   PROGRAM AREA                0014          20          3    00000640
   PUBLICATION DATE            0017          23          4    00000650
   TITLE                       001A          26          5    00000660
   PERSONAL AUTHOR             001B          27          6    00000670
   DESCRIPTOR                  0023          35          7    00000680
   IDENTIFIER                  0024          36          8    00000690
   E D R S  PRICE              0025          37          9    00000700
   DESCRIPTIVE NOTE            0026          38         10    00000710
   ISSUE                       0028          43         11    00000720
   ABSTRACT                    002C          44         12    00000730
   REPORT                      002D          45         13    00000740
   CONTRACT                    002E          46         14    00000750
   GRANT NUMBER                002F          47         15    00000760
   BUREAU NUMBER               0030          48         16    00000770
   AVAILABILITY                0031          49         17    00000780
   JOURNAL CITATION            0032          50         18    00000790
   INSTITUTION NAME            0080         128         19    00000800
   SPONSORING AGENCY           0084         132         20    00000810
  */                                                          00000820
  GENER: PROCEDURE OPTIONS(MAIN);                             00000830
  DCL A_OR_BLNK CHAR(1),                                      00000840
      #_DOC FIXED BINARY(15,0),                               00000850
     (ICODE, JCODE,KCODE) FIXED BINARY(15,0),                 00000860
      ISSUE CHAR(7);                                          00000870
                                                              00000880
   /*                                                         00000890
            INPUT CONTROL CARD FOR AIM OR ARM RECOGNITION,    00000900
            NUMBER OF DOCUMENTS, AND ISSUE                    00000910
    */                                                        00000920
                                                              00000930
      GET FILE(SYSIN) LIST(A_OR_BLNK,#_DOC,ISSUE);            00000940
      IF A_OR_BLNK='A' THEN JCODE=6; ELSE JCODE=9;            00000950
  BEGIN;                                                      00000960
      CLOSE FILE(SYSIN);                                      00000970
      DCL VTS(#_DOC) CHAR(10);                                00000980
     DCL CODE(#_DOC) CHAR(17),                                00000990
      BUF2 CHAR(2000) VAR,                                    00001000
           CARD CHAR(80);                                     00001010
  DCL                                                         00001020
      CODE_TABLE(20) FIXED BINARY(15,0),  /* WILL CONTAIN BIT CODES */ 00001030
      (I,CODES) FIXED BINARY (31,0),                          00001040
      LENGTHS FIXED BINARY (15,0),                            00001050
      TITLE CHAR(400) VAR,                                    00001060
      BUFFER CHAR(6996) VAR, /* TO HOLD INPUT STRING */       00001070
      OUT_STRING CHAR(2000) VAR;                              00001080
     DCL ACC_CODE CHAR(2) INITIAL(' &'),                      00001090
          BUFFER_LENGTH FIXED BINARY(31,0),                   00001100
     (J, J2, K, LEN_C_AUTHOR(5), LEN_P_AUTHOR(2), LEN_SA(3))  00001110
          FIXED BINARY(15,0),                                 00001120
      CHAR_LENGTH(17) FIXED BINARY(15,0),                     00001130
      #_P_AUTHORS          FIXED BINARY(15,0),                00001140
      TITLE_LEN           FIXED BINARY(15,0),                 00001150
      #_C_AUTHORS                                             00001160
                          FIXED BINARY(15,0),                 00001170
      #SA                                                     00001180
                          FIXED BINARY(15,0),                 00001190
      LEN_PRICE                                               00001200
                          FIXED BINARY(15,0),                 00001210
      LEN_CONTRACT                                            00001220
                          FIXED BINARY(15,0),                 00001230
      LEN_GRANT                                               00001240
```

```
                    FIXED BINARY(15,0),                                    00001250
      LEN_PA                                                               00001260
                    FIXED BINARY(15,0),                                    00001270
      LEN_REPORT                                                           00001280
                    FIXED BINARY(15,0),                                    00001290
      LEN_BUREAU                                                           00001300
                    FIXED BINARY(15,0),                                    00001310
      LEN_AVAIL                                                            00001320
                    FIXED BINARY(15,0),                                    00001330
      LEN_JC                                                               00001340
                    FIXED BINARY(15,0),                                    00001350
      LEN_DATE                                                             00001360
                    FIXED BINARY(15,0),                                    00001370
      LEN_PAGES                                                            00001380
                    FIXED BINARY(15,0),                                    00001390
      LEN_DESC_NOTE                                                        00001400
                    FIXED BINARY(15,0),                                    00001410
      LEN_DESC                                                             00001420
                    FIXED BINARY(15,0),                                    00001430
      LEN_IDENT                                                            00001440
                    FIXED BINARY(15,0),                                    00001450
      (BUREAU, PRICE, REPORT) CHAR(66) VAR,                                00001460
      (AVAIL, DESC_NOTE, JOURNAL, PERSONAL_AUTHOR) CHAR(132) VAR,          00001470
      (CORPORATE_AUTHOR, DESCRIPTOR) CHAR(660) VAR,                        00001480
      (CONTRACT, GRANT) CHAR(44) VAR,                                      00001490
      (IDENTIFIER, SPONSOR_AGENCY) CHAR(396) VAR,                          00001500
      DATE CHAR(7),                                                        00001510
      (VT, ED) CHAR(10),                                                   00001520
      PA CHAR(2) VAR,                                                      00001530
      PAGES CHAR(16) VAR,                                                  00001540
            NEW_CODE FIXED BINARY(15,0);                                   00001550
                                                                           00001560
   /*                                                                      00001570
            INITIALIZE CODE TABLE                                          00001580
   */                                                                      00001590
                                                                           00001600
        CODE_TABLE(1)=16;                                                  00001610
        CODE_TABLE(2)=17;                                                  00001620
       CODE_TABLE(3)=20;                                                   00001630
       CODE_TABLE(4)=23;                                                   00001640
       CODE_TABLE(5)=26;                                                   00001650
       CODE_TABLE(6)=27;                                                   00001660
    DO I=7 TO 10;                                                          00001670
       CODE_TABLE(I)=I+28;                                                 00001680
    END;                                                                   00001690
   DO I=11 TO 18;                                                          00001700
      CODE_TABLE(I)=32+I;                                                  00001710
    END;                                                                   00001720
       CODE_TABLE(19)=128;                                                 00001730
       CODE_TABLE(20)=132;                                                 00001740
                                                                           00001750
   /*                                                                      00001760
            E O F CONTROLS                                                 00001770
   */                                                                      00001780
                                                                           00001790
  ON ENDFILE(RIE) GO TO STOP;                                              00001800
  ON ENDFILE(VTSIN) GO TO INPUT;                                           00001810
        ON TRANSMIT(RIE) GO TO INPUT;                                      00001820
      KCODE=0;                                                             00001830
                                                                           00001840
   /*                                                                      00001850
            READ FILE VTSIN INTO FILE CARD AND GENERATE AN ORDERED         00001860
```

```
            SET OF VT NUMBERS WITH A CORRESPONDING SET OF ORDERED          00001870
            CODES                                                          00001880
   */                                                                      00001890
                                                                           00001900
LOOP: READ FILE(VTSIN) INTO(CARD);                                         00001910
      KCODE=KCODE+1;                                                       00001920
      VTS(KCODE)=SUBSTR(CARD,1,10);                                        00001930
      CODE(KCODE)=SUBSTR(CARD,15,17);                                      00001940
   PUT FILE(SYSPRINT) EDIT(KCODE,VTS(KCODE),CODE(KCODE))(X(5),F(4),X(2    00001950
      ),A(10),X(2),A(17));                                                 00001960
   GO TO LOOP;                                                             00001970
                                                                           00001980
   /*                                                                      00001990
            READ FILE RIE INTO BUFFER                                      00002000
   */                                                                      00002010
                                                                           00002020
INPUT: READ FILE(RIE) INTO(BUFFER);                                        00002030
                                                                           00002040
   /*                                                                      00002050
            INITIALIZE VARIABLES                                           00002060
   */                                                                      00002070
                                                                           00002080
CHAR_LENGTH=0;                                                             00002090
PERSONAL_AUTHOR,TITLE,CORPORATE_AUTHOR,SPONSOR_AGENCY,PRICE='';            00002100
CONTRACT,GRANT,REPORT,BUREAU,AVAIL,JOURNAL,DESC_NOTE,DESCRIPTOR='';        00002110
IDENTIFIER='';                                                             00002120
   DATE=' ';                                                               00002130
   PA,PAGES='';                                                            00002140
#_P_AUTHORS,TITLE_LEN,#_C_AUTHORS,#SA,LEN_PRICE,LEN_CONTRACT=0;            00002150
   LEN_AVAIL,LEN_JC,LEN_PAGES,LEN_DESC_NOTE,LEN_DESC,LEN_IDENT=0;          00002160
   LEN_DATE=7;                                                             00002170
LEN_GRANT,LEN_PA,LEN_REPORT,LEN_BUREAU=0;                                  00002180
LEN_P_AUTHOR=0;   LEN_C_AUTHOR=0;   LEN_SA=0;                              00002190
                                                                           00002200
   /*                                                                      00002210
            SCAN BUFFER FOR ITS LENGTH-BUFFER_LENGTH-, INITIALIZE -I-,     00002220
            SCAN BUFFER FOR THE LENGTH-CODE-OF THAT PORTION OF THE         00002230
            BUFFER THAT APPEARS BEFORE THE SUBSTRING ACC_CODE              00002240
   */                                                                      00002250
                                                                           00002260
   BUFFER_LENGTH=LENGTH(BUFFER);                                           00002270
   I=1;                                                                    00002280
CODES=INDEX(BUFFER,ACC_CODE);                                              00002290
                                                                           00002300
   /*                                                                      00002310
            DUMP INTO NEW_CODE THE FIELD CODE BIT STRING OF THE            00002320
            FIELD FROM THE RIE FILE BEING PROCESSED                        00002330
   */                                                                      00002340
                                                                           00002350
      UNSPEC(NEW_CODE)=UNSPEC(SUBSTR(BUFFER,CODES,2));                     00002360
                                                                           00002370
   /*                                                                      00002380
            IF THE FIELD CODE IS OUT OF POSITION, GO TO THE INPUT          00002390
            OF THE RIE FILE                                                00002400
   */                                                                      00002410
                                                                           00002420
      IF CODES<3 THEN GO TO INPUT;                                         00002430
                                                                           00002440
   /*                                                                      00002450
            DUMP INTO LENGTHS THE BIT STRING GIVING THE TOTAL LENGTH       00002460
            OF THE FIELD BEING PROCESSED                                   00002470
   */                                                                      00002480
```

```
 NEXT: UNSPEC(LENGTHS)=UNSPEC(SUBSTR(BUFFER,CODES-2,2));

    /*
          IF LENGTHS>BUFFER_LENGTH OR THE FIELD HAS NO DATA, WRITE
          LENGTHS, BUFFER_LENGTH, THEN RETURN TO THE INPUT OF THE
          RIE FILE
    */

      IF LENGTHS>BUFFER_LENGTH|LENGTHS<5 THEN DO;
            PUT FILE(SYSPRINT) EDIT(LENGTHS,BUFFER_LENGTH)(SKIP,
                2 F(4));
             GO TO INPUT;
                                        END;

    /*
          COPY DATA STRING FROM THE FIELD BEING PROCESSED AND STORE
          IT IN THE PROCESSING LOCATION DENOTED AS OUT_STRING
    */

          OUT_STRING=SUBSTR(BUFFER,CODES+2,LENGTHS-4);

    /*
          IF NEW_CODE=CODE_TABLE(I), THEN ENTER THE PROCESSING BLOCK
          AND STORE CHARACTER LENGTH AND LENGTH INDICATOR
    */

             IF NEW_CODE=CODE_TABLE(I) THEN
    DO;

    /*
          PROCESSING BLOCK:  PURPOSE:  GIVEN A CODE VALUE(I.E. -I-) THE
          BLOCK WILL CONSTRUCT THE APPROPRIATE STRING OF DATA FROM THE
          FIELD BEING PROCESSED.  ALONG WITH THE CONSTRUCTION OF NEW
          DATA STRINGS IN THEIR DESIRED NEW FORM, APPROPRIATE LENGTH
          PARAMETERS ARE SPECIFIED FOR EACH NEW STRING CONSTRUCTED.
          NOTE:  IF I=12, ABSTRACTS ARE BUILT, PRINTED OUT, AND STORED
          IN THE FILE VTABS.
    */

      IF I=1 THEN ED='ED '||SUBSTR(OUT_STRING,3,3)||' '||SUBSTR(
          OUT_STRING,6,3);
           ELSE
   IF I=2 THEN DO;
   VT=SUBSTR(OUT_STRING,1,2)||' '||SUBSTR(OUT_STRING,3,3)||' '||
     SUBSTR(OUT_STRING,6,3);
     IF SUBSTR(VT,1,2)¬='VT' THEN DO;
        PUT FILE(SYSPRINT) EDIT(BUFFER)(SKIP(2),A);
        GO TO INPUT;
                                END;
     DO ICODE=1 TO KCODE;
        IF VT=VTS(ICODE) THEN GO TO UPCODE;
     END;
     GO TO INPUT;
          END;
        ELSE
     IF I=3 THEN DO;
             PA=OUT_STRING;
             LEN_PA=LENGTHS-4;
             END;
        ELSE
   IF I=4&LENGTHS>5 THEN SUBSTR(DATE,12-LENGTHS)=OUT_STRING;
```

```
            ELSE                                                    00003110
         IF I=5 THEN DO;                                            00003120
        TITLE_LEN=LENGTHS-1;                                        00003130
         IF TITLE_LEN<5|TITLE_LEN>400 THEN DO;                      00003140
            PUT FILE(ERROR) EDIT (BUFFER)(SKIP,A);                  00003150
            GO TO INPUT;                        END;                00003160
     TITLE='<'||OUT_STRING||'>..';                                  00003170
                  END;                                              00003180
               ELSE                                                 00003190
         IF I=6 THEN DO;                                            00003200
                  J=INDEX(OUT_STRING,';');                          00003210
                     IF J=0 THEN DO;                                00003220
                        J=INDEX(OUT_STRING,'AND OTHERS');           00003230
            IF J=0 THEN J=INDEX(OUT_STRING,'AND OTHERS');           00003240
                     IF J=0 THEN DO;                                00003250
                        #_P_AUTHORS=2;                              00003260
            LEN_P_AUTHOR(1)=INDEX(OUT_STRING,';')-1;                00003270
         LEN_P_AUTHOR(2)=LENGTHS-6-LEN_P_AUTHOR(1);                 00003280
         PERSONAL_AUTHOR=SUBSTR(OUT_STRING,1,LEN_P_AUTHOR(1))||SUBSTR( 00003290
         OUT_STRING,LEN_P_AUTHOR(1)+3,LEN_P_AUTHOR(2));             00003300
                           END;                                     00003310
                     ELSE DO;                                       00003320
                        J=INDEX(OUT_STRING,';');                    00003330
                        SUBSTR(OUT_STRING,J,1)=',';                 00003340
                        #_P_AUTHORS=1;                              00003350
                        LEN_P_AUTHOR(1)=LENGTHS-4;                  00003360
                        PERSONAL_AUTHOR=OUT_STRING;                 00003370
                        END;                                        00003380
               END;                                                 00003390
                           ELSE DO;                                 00003400
                             #_P_AUTHORS=1;                         00003410
                             LEN_P_AUTHOR(1)=LENGTHS-4;             00003420
            PERSONAL_AUTHOR=OUT_STRING;                             00003430
                     END;                                           00003440
                  END;                                              00003450
            ELSE                                                    00003460
         IF I=7 THEN DO;                                            00003470
              DESCRIPTOR=OUT_STRING;                                00003480
                 LEN_DESC=LENGTHS-4;                                00003490
                                    END;                            00003500
               ELSE                                                 00003510
         IF I=8 THEN DO;                                            00003520
              IDENTIFIER=OUT_STRING;                                00003530
           LEN_IDENT=LENGTHS-4;                                     00003540
                  END;                                              00003550
             ELSE                                                   00003560
         IF I=9 THEN DO;                                            00003570
                     PRICE=OUT_STRING;                              00003580
                  LEN_PRICE=LENGTHS-4;                              00003590
                  END;                                              00003600
            ELSE                                                    00003610
         IF I=10 THEN DO;                                           00003620
               J=INDEX(OUT_STRING,'P.');                            00003630
            IF J<=0 THEN J=INDEX(OUT_STRING,'P.');                  00003640
                  IF J<LENGTHS-7 THEN DO;                           00003650
                    DESC_NOTE=SUBSTR(OUT_STRING,J+4);               00003660
                  LEN_DESC_NOTE=LENGTH(DESC_NOTE);                  00003670
                                    END;                            00003680
                  LEN_PAGES=J+1;                                    00003690
                                    PAGES=SUBSTR(OUT_STRING,1,J+1); 00003700
                  IF J<0 THEN DO;                                   00003710
                        LEN_DESC_NOTE=0;                            00003720
```

```
                        LEN_PAGES=0;                                    00003730
                        END;                                            00003740
                    END;                                                00003750
                ELSE                                                    00003760
    IF I=11 THEN GO TO UPCODE;                                          00003770
        ELSE                                                            00003780
    IF I=12 THEN DO;                                                    00003790
      BUF2=VT||SUBSTR(CODE(ICODE),JCODE,2)||OUT_STRING;                 00003800
    PUT FILE(SYSPRINT) EDIT(BUF2)(SKIP(3),A);                           00003810
    WRITE FILE(VTABS) FROM(BUF2);                                       00003820
                END;                                                    00003830
                ELSE                                                    00003840
        IF I=13 THEN DO;                                                00003850
                REPORT=OUT_STRING;                                      00003860
                  LEN_REPORT=LENGTHS-4;                                 00003870
                  END;                                                  00003880
                ELSE                                                    00003890
        IF I=14 THEN DO;                                                00003900
                CONTRACT=OUT_STRING;                                    00003910
                  LEN_CONTRACT=LENGTHS-4;                               00003920
                  END;                                                  00003930
          ELSE                                                          00003940
        IF I=15 THEN DO;                                                00003950
                GRANT=OUT_STRING;                                       00003960
                  LEN_GRANT=LENGTHS-4;                                  00003970
                  END;                                                  00003980
          ELSE                                                          00003990
        IF I=16 THEN DO;                                                00004000
                BUREAU=OUT_STRING;                                      00004010
                  LEN_BUREAU=LENGTHS-4;                                 00004020
                  END;                                                  00004030
          ELSE                                                          00004040
        IF I=17 THEN DO;                                                00004050
                AVAIL=OUT_STRING;                                       00004060
              LEN_AVAIL=LENGTH(AVAIL);                                  00004070
                  END;                                                  00004080
          ELSE                                                          00004090
        IF I=18 THEN DO;                                                00004100
                  LEN_JC=LENGTHS-4;                                     00004110
              JOURNAL=OUT_STRING;                                       00004120
                  END;                                                  00004130
          ELSE                                                          00004140
        IF I=19 THEN DO;                                                00004150
                  K=1;                                                  00004160
              CORPORATE_AUTHOR=OUT_STRING;                              00004170
                  J2=1;                                                 00004180
                  JLOOP: J=INDEX(CORPORATE_AUTHOR,';');                 00004190
                    IF J¬=0 THEN DO;                                    00004200
                      LEN_C_AUTHOR(K)=J-J2;                             00004210
                    K=K+1;                                              00004220
                      J2=J+1;                                           00004230
                      CORPORATE_AUTHOR=SUBSTR(CORPORATE_AUTHOR,1,J-     00004240
                      1)||SUBSTR(CORPORATE_AUTHOR,J+2);                 00004250
                      GO TO JLOOP;                                      00004260
                        END;                                            00004270
          ELSE IF J2=1 THEN LEN_C_AUTHOR(1)=LENGTH(OUT_STRING);         00004280
                    ELSE LEN_C_AUTHOR(K)=LENGTH(OUT_STRING)-J2;         00004290
                      #_C_AUTHORS=K;                                    00004300
                  END;                                                  00004310
                ELSE                                                    00004320
        IF I=20 THEN DO;                                                00004330
                  K=1;                                                  00004340
```

```
                SPONSOR_AGENCY=OUT_STRING;                                      00004350
                 J2=1;                                                          00004360
              JSALOOP: J=INDEX(SPONSOR_AGENCY,';');                             00004370
                       IF J¬=0 THEN DO;                                         00004380
                         LEN_SA(K)=J-J2;                                        00004390
                         K=K+1;                                                 00004400
                         J2=J+1;                                                00004410
                         SPONSOR_AGENCY=SUBSTR(SPONSOR_AGENCY,1,J-1)||          00004420
                           SUBSTR(SPONSOR_AGENCY,J+2);                          00004430
                         GO TO JSALOOP;                                         00004440
                                  END;                                          00004450
           ELSE IF J2=1 THEN LEN_SA(1)=LENGTH(OUT_STRING);                      00004460
                        ELSE LEN_SA(K)=LENGTH(OUT_STRING)-J2;                   00004470
                       #SA=K;                                                   00004480
                 END;                                                           00004490
   END;                                                                         00004500
                                                                                00004510
    /*                                                                          00004520
          TO CODES, THE LENGTH OF THE FIELD , BEING PROCESSED, THAT             00004530
          APPEARS BEFORE THE SUBSTRING ACC_CODE, ADD THE LENGTH OF              00004540
          THE ENTIRE FIELD. THIS IS TO BE USED AS A CHECK FOR TERMINAT          00004550
          ING THE PROCESSING OF A COMPLETE RECORD.                              00004560
    */                                                                          00004570
                                                                                00004580
  UPCODE: CODES=CODES+LENGTHS;                                                  00004590
                                                                                00004600
    /*                                                                          00004610
          IF CODES>BUFFER_LENGTH, AN ENTIRE RECORD HAS BEEN PROCESSED.          00004620
          PERFORM OUTPUT AND STORAGE, THEN GO TO INPUT OF THE RIE FILE          00004630
    */                                                                          00004640
                                                                                00004650
    IF CODES>BUFFER_LENGTH|I=20 THEN DO;                                        00004660
          PUT STRING(BUFFER) EDIT(VT,CODE(ICODE),ED,ISSUE,                      00004670
  PERSONAL_AUTHOR,TITLE,CORPORATE_AUTHOR,SPONSOR_AGENCY,PRICE,CONTRACT,         00004680
  GRANT,PA,REPORT,BUREAU,AVAIL,JOURNAL,DATE,PAGES,DESC_NOTE,DESCRIPTOR,         00004690
  IDENTIFIER,'||',#_P_AUTHORS,(LEN_P_AUTHOR(J) DO J=1 TO #_P_AUTHORS),          00004700
  TITLE_LEN,#_C_AUTHORS,(LEN_C_AUTHOR(J) DO J=1 TO #_C_AUTHORS),#SA,            00004710
  (LEN_SA(J) DO J=1 TO #SA),LEN_PRICE,LEN_CONTRACT,LEN_GRANT,LEN_PA,            00004720
  LEN_REPORT,LEN_BUREAU,LEN_AVAIL,LEN_JC,LEN_DATE,LEN_PAGES,LEN_DESC_NOTE00004730
  ,LEN_DESC,LEN_IDENT)(A(10),A(17),A(20),A(7),X(4),18 A,28 F(4));               00004740
    PUT FILE(SYSPRINT) EDIT(BUFFER)(SKIP,A);                                    00004750
   WRITE FILE(WORK) FROM(BUFFER);                                               00004760
    GO TO INPUT;                                                                00004770
                                  END;                                          00004780
                                                                                00004790
    /*                                                                          00004800
          DUMP INTO NEW_CODE THE FIELD CODE -ACC_CODE- BIT STRING               00004810
    */                                                                          00004820
                                                                                00004830
       UNSPEC(NEW_CODE)=UNSPEC(SUBSTR(BUFFER,CODES,2));                         00004840
                                                                                00004850
    /*                                                                          00004860
          SCAN CODE_TABLE                                                       00004870
          FOR VALUES OF I FROM 2 TO 20 CHECK IF NEW=CODE_TABLE(I).              00004880
          IF A MATCH IS FOUND , GO TO STATEMENT NAMED NEXT.                     00004890
          IF FOR A VALUE OF I, NEW_CODE<CODE_TABLE(I), SET I=I-1                00004900
          AND GO TO NEXT.                                                       00004910
          OTHERWISE, GO TO INPUT OF THE RIE FILE.                               00004920
    */                                                                          00004930
                                                                                00004940
       DO I=I+1 TO 20;                                                          00004950
                IF NEW_CODE=CODE_TABLE(I) THEN GO TO NEXT;                      00004960
```

```
            IF NEW_CODE<CODE_TABLE(I) THEN DO;                          00004970
                 I=I-1;                                                 00004980
                 GO TO NEXT;                                            00004990
                                        END;                            00005000
            END;                                                        00005010
          GO TO INPUT;                                                  00005020
      STOP: CLOSE FILE(WORK), FILE(VTABS),FILE(RIE);                    00005030
      END GENER;                                                        00005040
```

MASTER

```
/*************************************************************************/00000010
/*                          M A S T E R   P R O G R A M                  */00000020
/*                                                                       */00000040
/*                                                                       */00000050
/*                                                                       */
/*  PURPOSE:  THIS PROGRAM GENERATES A MASTER TAPE DATA SET FOR          */00000060
/*            THE QUARTERLY AIM OR ARM DATA.  THE PROGRAM CHECKS         */00000070
/*            THE DATA AS THOROUGHLY AS POSSIBLE FOR ERRORS.  IF         */00000080
/*            THE DATA FOR A DOCUMENT IS CORRECT, A RECORD IS FORMED     */00000090
/*            FOR THE DOCUMENT ON THE MASTER TAPE.  IF THERE ARE         */00000100
/*            ERRORS FLAGGED, AN ERROR MESSAGE IS PRINTED OUT AND NO     */00000110
/*            RECORD GOES ON THE MASTER TAPE.  WHEN ALL THE ERRORS       */00000120
/*            HAVE BEEN FLAGGED, CORRECT THE BAD DATA.  THEN CHANGE      */00000130
/*            THE DISPOSITION ON THE MASTER DD CARD TO  -MOD-  AND       */00000140
/*            THESE DOCUMENTS WILL HAVE THEIR RECORDS ADDED ON TO        */00000150
/*            THE END OF THE DATA SET.                                   */00000160
/*                                                                       */00000170
/*  DEFINITION OF VARIABLES:                                             */00000180
/*     1. NA -- MAXIMUM NUMBER OF PERSONAL AUTHORS THAT CAN BE FOUND     */00000190
/*           IN THAT SECTION.  (CURRENTLY SET AT 2)                      */00000200
/*     2. N10 -- MAXIMUM LINE NUMBER FOUND IN THE PERSONAL AUTHOR        */00000210
/*           SECTION. (CURRENTLY SET AT 10)                              */00000220
/*     3. NT  -- MAXIMUM LINE NUMBER FOUND IN THE TITLE SECTION.         */00000230
/*           (CURRENTLY SET AT 22)                                       */00000240
/*     4. NC -- MAXIMUM LINE NUMBER FOUND IN THE CORPORATE AUTHOR        */00000250
/*           SECTION. (CURRENTLY SET AT 34).                             */00000260
/*     5. NS -- MAXIMUM LINE NUMBER FOUND IN THE SPONSORING AGENCY       */00000270
/*           SECTION. (CURRENTLY SET AT 42)                              */00000280
/*     6. N50 -- MAXIMUM LINE NUMBER FOUND IN THE PRICE SECTION.         */00000290
/*           (CURRENTLY SET AT 50)                                       */00000300
/*     7. N60 -- MAXIMUM LINE NUMBER FOUND IN THE REPORT OR BUREAU       */00000310
/*           NUMBER SECTION. (CURRENTLY SET AT 60)                       */00000320
/*     8. N70 -- MAXIMUM LINE NUMBER FOUND IN THE AVAILABILITY           */00000330
/*           SECTION. (CURRENTLY SET AT 71)                              */00000340
/*     9. N80 -- MAXIMUM LINE NUMBER FOUND IN THE JOURNAL CITATION       */00000350
/*           SECTION. (CURRENTLY SET AT 80)                              */00000360
/*    10. N90 -- MAXIMUM LINE NUMBER FOUND IN THE DATE, PAGE, AND        */00000370
/*           DESCRIPTIVE NOTE SECTION. (CURRENTLY SET AT 91)             */00000380
/*    11. N100 -- MAXIMUM LINE NUMBER FOUND IN THE DESCRIPTOR            */00000390
/*           SECTION. (CURRENTLY SET AT 104)                             */00000400
/*    12. N200 -- MAXIMUM LINE NUMBER FOUND IN THE IDENTIFIER            */00000410
/*           SECTION. (CURRENTLY SET AT 202)                             */00000420
/*    13. NDOC -- THE NUMBER OF DOCUMENTS BEING PROCESSED BY THE         */00000430
/*           MASTER PROGRAM FOR THIS ISSUE OF AIM OR ARM.                */00000440
/*    14. AUTHOR -- THE NAME OF A PERSONAL AUTHOR.                       */00000450
/*    15. AVAILABILITY -- THE AVAILABILITY STRING.                       */00000460
/*    16. BD -- THE CHARACTER FOUND IN COLUMN 1 OF THE CARD THAT         */00000470
/*           SEPARATES THE DOCUMENT DATA ON CARDS.  (¬)                  */00000480
/*    17. BT -- CHARACTER DENOTING BEGIN SHORT TITLE. (<)                */00000490
/*    18. BUREAU_NO -- THE BUREAU NUMBER.                                */00000500
/*    19. CARD_NO -- CARD IDENTIFICATION NUMBER FOUND IN COLUMN 14       */00000510
/*           OF THE DATA CARDS.                                          */00000520
/*    20. CAT -- STRING WITH ALL THE DOCUMENT'S DATA USED TO MAKE        */00000530
/*           THE RECORD ON TAPE.                                         */00000540
/*    21. CODE_NO -- CODE NUMBER.                                        */00000550
/*    22. CONTRACT_NO -- CONTRACT NUMBER.                                */00000560
/*    23. CORP_AUTHOR -- CORPORATE AUTHOR.                               */00000570
/*    24. CR -- CHARACTER DENOTING END OF INFORMATION ON DATA CARDS.     */00000580
/*           (|)                                                         */00000590
/*    25. DESCRIP -- STRING OF DESCRIPTORS.                              */00000600
/*    26. DESC_NOTE -- DESCRIPTIVE NOTE.                                 */00000610
/*    27. E -- ARRAY OF ED OR ACCESSION NUMBERS FOR THE DOCUMENTS        */00000620
/*           BEING PROCESSED.                                            */00000630
```

```
/*    28. ED_NO -- ACCESSION NUMBER.                                    */00000640
/*    29. ET -- CHARACTER DENOTING END SHORT TITLE. (>)                 */00000650
/*    30. GRANT_NO -- GRANT NUMBER.                                     */00000660
/*    31. IDENT -- STRING OF IDENTIFIERS.                               */00000670
/*    32. IND -- INDICATOR WHICH SAYS WHETHER WE WERE AT THE END        */00000680
/*              OF A DOCUMENT OR NOT WHEN WE HIT AN UNMATCHING          */00000690
/*              VT NUMBER. WHEN WE HIT THE BEGIN DOCUMENT CARD,         */00000700
/*              THERE WILL BE A MISMATCHED VT NUMBER FLAGGED. BUT       */00000710
/*              THERE IS NO REAL ERROR. -IND- TELLS US THIS.            */00000720
/*    33. IMP -- HELPS POSITION US AT THE BEGINNING OF A NEW            */00000730
/*              DOCUMENT WHEN THE PREVIOUS ONE IS FINISHED.             */00000740
/*    34. JOUR_CIT -- JOURNAL CITATION.                                 */00000750
/*    35. LINE_NO -- LINE NUMBER ON THE DATA CARD.                      */00000760
/*    36. LONG -- INTERMEDIATE CHARACTER STRING.                        */00000770
/*    37. L_AUTHOR -- LENGTH OF AN AUTHOR'S NAME.                       */00000780
/*    38. L_AVAILABILITY -- LENGTH OF AVAILABILITY STRING.              */00000790
/*    39. L_BUREAU_NO -- LENGTH OF BUREAU NUMBER.                       */00000800
/*    40. L_CONTRACT_NO -- LENGTH OF CONTRACT NUMBER.                   */00000810
/*    41. L_CORP_AUTHOR -- LENGTH OF A CORPORATE AUTHOR NAME.           */00000820
/*    42. L_DESCRIP -- LENGTH OF DESCRIPTOR STRING.                     */00000830
/*    43. L_DESC_NOTE -- LENGTH OF DESCRIPTIVE NOTE.                    */00000840
/*    44. L_GRANT_NO -- LENGTH OF GRANT NUMBER.                         */00000850
/*    45. L_IDENT -- LENGTH OF IDENTIFIER STRING.                       */00000860
/*    46. L_JOUR_CIT -- LENGTH OF JOURNAL CITATION.                     */00000870
/*    47. L_PAGE -- LENGTH OF PAGE STRING.                              */00000880
/*    48. L_PA_NO -- LENGTH OF PA NUMBER.                               */00000890
/*    49. L_PRICE -- LENGTH OF PRICE.                                   */00000900
/*    50. L_PUB_DATE -- LENGTH OF PUBLICATION DATE.                     */00000910
/*    51. L_REPORT_NO -- LENGTH OF REPORT NUMBER.                       */00000920
/*    52. L_SPONSOR_AG -- LENGTH OF A SPONSORING AGENCY.                */00000930
/*    53. L_TITLE -- LENGTH OF TITLE.                                   */00000940
/*    54. MASTER -- NAME OF MASTER TAPE.                                */00000950
/*    55. NO_AUTHOR -- ACTUAL NUMBER OF PERSONAL AUTHORS FOR A          */00000960
/*                     DOCUMENT.                                        */00000970
/*    56. NO_CORP_AUTHOR -- ACTUAL NUMBER OF CORPORATE AUTHORS FOR A    */00000980
/*                     DOCUMENT.                                        */00000990
/*    57. NO_DES -- FLAG TO INDICATE IF THERE ARE ANY IDENTIFIERS       */00001000
/*                  FOR A DOCUMENT.                                     */00001010
/*    58. NO_RECS -- COUNTS THE NUMBER OF RECORDS ON THE MASTER TAPE.   */00001020
/*    59. NO_SPONSOR -- ACTUAL NUMBER OF SPONSORING AGENCIES FOR A      */00001030
/*                     DOCUMENT.                                        */00001040
/*    60. PAGE -- NUMBER OF PAGES IN THE REPORT.                        */00001050
/*    61. PAGE_NO -- PAGE NUMBER IN THE AIM OR ARM PUBLICATION ON       */00001060
/*                   WHICH THE ABSTRACT OF THE DOCUMENT CAN BE FOUND.   */00001070
/*                                                                      */00001080
/*    62. PA_NO -- PA NUMBER.                                           */00001090
/*    63. PG -- ARRAY OF PAGE NUMBERS IN THE PUBLICATION ON WHICH THE   */00001100
/*              ABSTRACTS CAN BE FOUND.                                 */00001110
/*    64. PRICE -- PRICE STRING.                                        */00001120
/*    65. PUB_DATE -- PUBLICATION DATE.                                 */00001130
/*    66. REPORT_NO -- REPORT NUMBER.                                   */00001140
/*    67. SEASON -- INDICATES AIM OR ARM AND FOR WHAT SEASON THE        */00001150
/*                  PUBLICATION IS BEING PRINTED.                       */00001160
/*    68. SEP -- SEPARATER USED IN THE MASTER RECORD (|).               */00001170
/*    69. SPONSOR_AG -- NAME OF A SPONSORING AGENCY.                    */00001180
/*    70. STR -- STRING INTO WHICH THE CARD DATA IS READ.               */00001190
/*    71. TAB -- INDICATES WHETHER A CARD SHOULD BE READ OR NOT.        */00001200
/*    72. TB -- TAB SEPARATING DATA WHEN MORE THAN ONE PIECE OF DATA    */00001210
/*              APPEARS ON THE SAME LINE. (¬)                           */00001220
/*    73. TITLE -- TITLE OF A DOCUMENT.                                 */00001230
/*    74. V -- ARRAY OF VT OR DOCUMENT NUMBERS BEING PROCESSED.         */00001240
/*    75. VT -- DOCUMENT NUMBER ON FIRST CARD OF DATA FOR A             */00001250
/*              PARTICULAR DOCUMENT.                                    */
```

```
/*     76. VT_NO -- DOCUMENT NUMBER ON SUCCEEDING DATA CARDS FOR       */00001260
/*                  A DOCUMENT.                                        */00001270
/*     77. YEAR -- CURRENT YEAR.                                       */00001280
/*                                                                     */00001290
/*  INPUT TO PROGRAM:                                                  */00001300
/*    -SYSIN-, INPUT ON CARDS:                                         */00001310
/*      CARD 1 -- CONTAINS VALUES FOR THE FOLLOWING VARIABLES, ALL IN  */00001320
/*                F(3,0) FIELDS: NA,N10,NT,NC,NS,N50,N60,N70,N80,N90,  */00001330
/*                N100,N200. THESE VARIABLES ARE DEFINED IN THE        */00001340
/*                PRECEDING SECTION AND THEIR CURRENT VALUES ARE LIST  */00001350
/*                AS WELL. CHANGES IN THESE NUMBERS WILL COME THROUGH  */00001360
/*                THE VOCATIONAL CENTER.                               */00001370
/*      CARD 2 -- CONTAINS THE NUMBER OF DOCUMENTS IN THIS QUARTERLY   */00001380
/*                DATA BEING PROCESSED. THE CENTER CAN SUPPLY THE      */00001390
/*                VALUE. RIGHT JUSTIFY IN COL 1-5.                     */00001400
/*      CARD 3 -- COL 1 IS THE ISSUE, 'A' FOR AIM OR ' ' FOR ARM.      */00001410
/*                COL 2-3 IS THE SEASON, 'F ' FOR FALL, 'W ' FOR       */00001420
/*                  WINTER, 'SP' FOR SPRING, 'SU' FOR SUMMER.          */00001430
/*                COL 4-7 IS THE CURRENT YEAR, 19__.                   */00001440
/*      CARDS 4-? -- THESE ARE THE VT, ED, RESUME PAGE CARDS. THE      */00001450
/*                   FORMAT FOR EACH CARD IS:                          */00001460
/*                   COL 1-10 -- VT OR DOCUMENT NUMBER.                */00001470
/*                   COL 11 - 30 -- ED OR ACCESSION NUMBER.            */00001480
/*                   COL 31 - 34 -- RESUME PAGE.                       */00001490
/*                   THE LAST CARD IN THIS DECK MUST BE A BLANK CARD.  */00001500
/*    -CARDS-, INPUT ON TAPE OR CARDS:                                 */00001510
/*      THE FIRST RUN WITH A NEW SET OF DATA, -CARDS- WILL BE A DD     */00001520
/*      CARD DEFINING THE CARD IMAGE TAPE. MASTER WILL PROBABLY FLAG   */00001530
/*      THE BAD DOCUMENTS. CORRECT THE DATA CARDS IN ERROR FOR THESE   */00001540
/*      DOCUMENTS. THEN RERUN THE PROGRAM WITH JUST THE DATA CARDS     */00001550
/*      FOR THE FLAGGED DOCUMENTS FOLLOWING A -//CARDS DD *- CARD.     */00001560
/*      MAKE SURE THE MASTER DD CARD HAS A DISPOSITION OF -MOD-. THE   */00001570
/*      BAD DOCUMENTS WILL THEN BE ADDED TO THE MASTER RECORDS         */00001580
/*      ALREADY ON TAPE.                                               */00001590
/*                                                                     */00001600
/*  SPECIAL NOTE:                                                      */00001610
/*    IT WILL BE IMPOSSIBLE TO UNDERSTAND THIS MASTER PROGRAM UNLESS   */00001620
/*    YOU ARE FAMILIAR WITH 3 THINGS:                                  */00001630
/*    (1) THE FORMAT OF THE MASTER RECORD                              */00001640
/*    (2) THE FORMAT OF THE DATA CARDS                                 */00001650
/*    (3) THE NATURE OF THE DATA ON THE CARDS                          */00001660
/*                                                                     */00001670
/*******************************************************************/00001680
                                                                       00001690
                                                                       00001700
BEGIN: PROCEDURE OPTIONS (MAIN);                                       00001710
          DECLARE                                                      00001720
            DUMMY                                                      00001730
                CHARACTER (1),                                         00001740
            (NA,N10,NT,NC,NS,N50,N60,N70,N80,N90,N100,N200)            00001750
                FIXED BINARY (31,0),                                   00001760
            NDOC                FIXED BINARY (31,0),                   00001770
          ;                                                            00001780
                                                                       00001790
                                                                       00001800
    /*  READ IN SOME MAXIMUM BOUNDARIES FOUND ON THE DATA CARDS.  */   00001810
                                                                       00001820
                                                                       00001830
          GET FILE (SYSIN) EDIT (NA,N10,NT,NC,NS,N50,N60,N70,N80,N90,  00001840
              N100,N200,DUMMY) (12 F(3),X(43),A(1));                   00001850
          GET FILE (SYSIN) EDIT (NDOC,DUMMY) (F(5),X(74),A(1));        00001860
                                                                       00001870
```

```
WORK:   BEGIN:                                                          00001880
                                                                        00001890
        DECLARE                                                         00001900
          AUTHOR(NA)          CHARACTER (50) VARYING,                   00001910
          AVAILABILITY        CHARACTER ((N70-69)*132) VARYING,         00001920
          BD                  CHARACTER (1) INITIAL ('¬'),              00001930
          BEG                                                           00001940
              CHARACTER (1),                                            00001950
          BT                  CHARACTER (1) INITIAL ('<'),              00001960
          BUREAU_NO           CHARACTER ((N60-59)*132/2) VARYING,       00001970
          CARD_NO                                                       00001980
              FIXED BINARY (31,0),                                      00001990
          CAT                 CHARACTER (3736) VARYING,                 00002000
          CODE_NO                                                       00002010
              CHARACTER (17),                                           00002020
          CONTRACT_NO         CHARACTER ((N50-49)*132/3) VARYING,       00002030
          CORP_AUTHOR(NC-29) CHARACTER (132) VARYING,                   00002040
          CR                  CHARACTER (1) INITIAL ('|'),              00002050
          DESCRIP             CHARACTER ((N100-99)*132) VARYING,        00002060
          DESC_NOTE           CHARACTER ((N90-89)*132-13) VARYING,      00002070
          E(NDOC)             CHARACTER (20),                           00002080
          ED_NO               CHARACTER (20),                           00002090
          ET                  CHARACTER (1) INITIAL ('>'),              00002100
          GRANT_NO            CHARACTER ((N50-49)*132/3) VARYING,       00002110
          I                                                             00002120
              FIXED BINARY (31,0),                                      00002130
          IDENT               CHARACTER ((N200-199)*132) VARYING,       00002140
          IMP                 FIXED BINARY (31,0),                      00002150
          IND                 FIXED BINARY (31,0),                      00002160
          JOUR_CIT            CHARACTER ((N80-79)*132) VARYING,         00002170
          K                                                             00002180
              FIXED BINARY (31,0),                                      00002190
          L                                                             00002200
              FIXED BINARY (31,0),                                      00002210
          LINE_NO                                                       00002220
              FIXED BINARY (31,0),                                      00002230
          LONG                CHARACTER (660) VARYING,                  00002240
          L_AUTHOR(NA)        FIXED BINARY (31,0),                      00002250
          L_AVAILABILITY                                                00002260
              FIXED BINARY (31,0),                                      00002270
          L_BUREAU_NO                                                   00002280
              FIXED BINARY (31,0),                                      00002290
          L_CONTRACT_NO                                                 00002300
              FIXED BINARY (31,0),                                      00002310
          L_CORP_AUTHOR(NC-29) FIXED BINARY (31,0),                     00002320
          L_DESCRIP                                                     00002330
              FIXED BINARY (31,0),                                      00002340
          L_DESC_NOTE                                                   00002350
              FIXED BINARY (31,0),                                      00002360
          L_GRANT_NO                                                    00002370
              FIXED BINARY (31,0),                                      00002380
          L_IDENT                                                       00002390
              FIXED BINARY (31,0),                                      00002400
          L_JOUR_CIT                                                    00002410
              FIXED BINARY (31,0),                                      00002420
          L_PAGE                                                        00002430
              FIXED BINARY (31,0),                                      00002440
          L_PA_NO                                                       00002450
              FIXED BINARY (31,0),                                      00002460
          L_PRICE                                                       00002470
              FIXED BINARY (31,0),                                      00002480
          L_PUB_DATE                                                    00002490
```

```
        FIXED BINARY (31,0),                                          00002500
    L_REPORT_NO                                                       00002510
        FIXED BINARY (31,0),                                          00002520
    L_SPONSOR_AG(NS-39) FIXED BINARY (31,0),                          00002530
    L_TITLE                                                           00002540
        FIXED BINARY (31,0),                                          00002550
    MASTER                                                            00002560
        FILE RECORD OUTPUT SEQUENTIAL UNBUFFERED,                     00002570
    N                                                                 00002580
        FIXED BINARY (31,0),                                          00002590
    NO_AUTHOR                                                         00002600
        FIXED BINARY (31,0),                                          00002610
    NO_CORP_AUTHOR                                                    00002620
        FIXED BINARY (31,0),                                          00002630
    NO_DES                                                            00002640
        FIXED BINARY (31,0),                                          00002650
    NO_RECS                                                           00002660
        FIXED BINARY (31,0),                                          00002670
    NO_SPONSOR                                                        00002680
        FIXED BINARY (31,0),                                          00002690
    PAGE                                                              00002700
        CHARACTER (16) VARYING,                                       00002710
    PAGE_NO          CHARACTER (4),                                   00002720
    PA_NO                                                             00002730
        CHARACTER (2),                                                00002740
    PG(NDOC)         CHARACTER (4),                                   00002750
    PRICE            CHARACTER ((N50-49)*132/2) VARYING,              00002760
    PUB_DATE                                                          00002770
        CHARACTER (7),                                                00002780
    REPORT_NO        CHARACTER ((N60-59)*132/2) VARYING,              00002790
    SEASON           CHARACTER (3),                                   00002800
    SEP              CHARACTER (2) INITIAL ('||'),                    00002810
    SEPERATE                                                          00002820
        ENTRY (FIXED BINARY (31,0),FIXED BINARY (31,0),               00002830
          FIXED BINARY (31,0),,),                                     00002840
    SPONSOR_AG(NS-39) CHARACTER (132) VARYING,                        00002850
    STR              CHARACTER (80),                                  00002860
    TAB                                                               00002870
        FIXED BINARY (31,0),                                          00002880
    TB               CHARACTER (1) INITIAL ('¬'),                     00002890
    TITLE            CHARACTER ((NT-19)*132) VARYING,                 00002900
    V(NDOC)          CHARACTER (10),                                  00002910
    VT                                                                00002920
        CHARACTER (10),                                               00002930
    VT_NO                                                             00002940
        CHARACTER (10),                                               00002950
    YEAR                                                              00002960
        CHARACTER (4),                                                00002970
    ;                                                                 00002980
                                                                      00002990
  OPEN FILE (ERRE) PRINT STREAM OUTPUT LINESIZE(125),                 00003000
       FILE (CARDS) INPUT RECORD,                                     00003010
       FILE (MASTER);                                                 00003020
 ON ENDFILE (CARDS) GO TO STOP;                                       00003030
 ON CONVERSION BEGIN;                                                 00003040
   PUT FILE (ERRE) EDIT ('CONVERSION ERROR IN DOCUMENT ',VT)          00003050
     (SKIP(2),A(29),A(10));                                           00003060
   GO TO SKIP;                                                        00003070
 END;                                                                 00003080
                                                                      00003090
                                                                      00003100
/*  READ IN REMAINING SYSIN CARDS.  */                                00003110
```

```
          GET FILE (SYSIN) EDIT (SEASON,YEAR,DUMMY)                          00003120
              (A(3),A(4),X(72),A(1));                                        00003130
ED_LP:    DO I = 1 TO NDOC;                                                  00003140
             GET FILE (SYSIN) EDIT (V(I),E(I),PG(I),DUMMY)                   00003150
                 (A(10),A(20),A(4),X(45),A(1));                              00003160
             IF V(I) = '          ' THEN GO TO ASGN;                         00003170
             K = I;                                                          00003180
          END ED_LP;                                                         00003190
ASGN:     NO_RECS = 0;                                                       00003200
          READ FILE (CARDS) IGNORE (1);                                      00003210
                                                                             00003220
   /*  READ IN LINE 1 WITH THE CODE NUMBER. CHECK FOR ERRORS INCLUDING A     00003230
       TO MAKE SURE THE SUB-CODE IS WITHIN BOUNDS. ANY ERRORS FOUND IN A     00003240
       DOCUMENT THROUGHOUT THIS PROGRAM WILL CAUSE AN ERROR MESSAGE TO BE    00003250
       PRINTED AND A SKIP TO THE BEGINNING OF THE NEXT DOCUMENT.  */         00003260
                                                                             00003270
START:    READ FILE (CARDS) INTO (STR);                                      00003280
          GET STRING (STR) EDIT (VT_NO,LINE_NO,CARD_NO,CODE_NO)              00003290
             (A(10),F(3),F(1),A(17));                                        00003300
          IND = 0;                                                           00003310
          IMP = 2;                                                           00003320
          TAB = 0;                                                           00003330
          VT = VT_NO;                                                        00003340
          N = INDEX(CODE_NO,CR);                                             00003350
          IF N = 0 THEN GO TO TEST;                                          00003360
          CODE_NO = SUBSTR(CODE_NO,1,N-1);                                   00003370
TEST:     IF LINE_NO ¬= 1 THEN GO TO ERR;                                    00003380
          IF CARD_NO ¬= 1 THEN GO TO ERR;                                    00003390
          IF (SUBSTR(CODE_NO,6,2)<40) | (SUBSTR(CODE_NO,6,2)>52)            00003400
            THEN DO;                                                         00003410
              PUT FILE (SYSPRINT) EDIT ('CODE NO. IS OUT OF BOUNDS',VT)      00003420
                (SKIP(2),A(25),A(10));                                       00003430
              GO TO SKIP;                                                    00003440
              END;                                                           00003450
                                                                             00003460
   /*  CHECK FOR ANY PERSONAL AUTHORS AND STORE THEIR NAMES IN -AUTHOR-  */  00003470
                                                                             00003480
          NO_AUTHOR = 0;                                                     00003490
          CALL SEPERATE(10,N10,1,TLE_SECT,LONG);                             00003500
SRCH:     N = INDEX(LONG,';');                                               00003510
          IF N ¬= 0 THEN DO;                                                 00003520
                         NO_AUTHOR = NO_AUTHOR + 1;                          00003530
                         IF NO_AUTHOR > NA THEN GO TO OFF_LNTH;              00003540
                         AUTHOR(NO_AUTHOR) = SUBSTR(LONG,1,N-1);             00003550
                         L_AUTHOR(NO_AUTHOR) = N - 1;                        00003560
                         IF (N-1) > 50 THEN GO TO OFF_LNTH;                  00003570
                         LONG = SUBSTR(LONG,N+2);                            00003580
                         GO TO SRCH;                                         00003590
                         END;                                                00003600
          NO_AUTHOR = NO_AUTHOR + 1;                                         00003610
          IF NO_AUTHOR > NA THEN GO TO OFF_LNTH;                             00003620
          N = INDEX(LONG,CR);                                                00003630
          IF N = 0 THEN GO TO MISSCR;                                        00003640
          AUTHOR(NO_AUTHOR) = SUBSTR(LONG,1,N-1);                            00003650
          L_AUTHOR(NO_AUTHOR) = N - 1;                                       00003660
          IF (N-1) > 50 THEN GO TO OFF_LNTH;                                 00003670
                                                                             00003680
                                                                             00003690
                                                                             00003700
                                                                             00003710
                                                                             00003720
                                                                             00003730
```

```
/*  PULL THE TITLE OFF OF THE DATA CARDS. CHECK FOR AN EVEN NUMBER OF
    AND END SHORT TITLE MARKS IN THE TITLE.  */

TLE_SECT:
          CALL SEPERATE (20,NT,2,CORP_SECT,LONG);
          N = INDEX(LONG,CR);
          IF N = 0 THEN GO TO MISSCR;
          TITLE = SUBSTR(LONG,1,N-1);
          L_TITLE = N - 1;
          I = 0;
          IDENT = TITLE;
RET:      N = INDEX(IDENT,BT);
          L = INDEX(IDENT,ET);
          IF (N=0) & (L=0) & (I¬=0) THEN GO TO CORP_SECT;
          IF (N=0) | (L=0)
            THEN DO:
                 PUT FILE (ERRE) EDIT
                   ('SHORT TITLE MARKS ARE NOT CORRECT IN DOCUMENT ',VT)
                   (SKIP(2),A(46),A(10));
                 GO TO SKIP;
                 END;
          I = I + 1;
          IDENT = SUBSTR(IDENT,L+2);
          GO TO RET;

/*  CHECK FOR ANY CORPORATE AUTHORS AND STORE NAMES AND NAME LENGTHS I
    ARE FOUND.  */


CORP_SECT:
          NO_CORP_AUTHOR = 0;
CP_LP:    DO I = 30 TO NC;
          CALL SEPERATE (I,I,3,SPON_SECT,LONG);
          NO_CORP_AUTHOR = NO_CORP_AUTHOR + 1;
          N = INDEX(LONG,CR);
          IF N = 0 THEN GO TO MISSCR;
          CORP_AUTHOR(NO_CORP_AUTHOR) = SUBSTR(LONG,1,N-1);
          L_CORP_AUTHOR(NO_CORP_AUTHOR) = N - 1;
          IF (N-1) > 132 THEN GO TO OFF_LNTH;
          END CP_LP;

/*  CHECK FOR ANY SPONSORING AGENCIES AND STORE NAMES AND NAME LENGTHS
    ARE FOUND.  */


SPON_SECT:
          NO_SPONSOR = 0;
SP_LP:    DO I = 40 TO NS;
          CALL SEPERATE (I,I,4,PRICE_SECT,LONG);
          NO_SPONSOR=NO_SPONSOR + 1;
          N = INDEX(LONG,CR);
          IF N = 0 THEN GO TO MISSCR;
          SPONSOR_AG(NO_SPONSOR) = SUBSTR(LONG,1,N-1);
          L_SPONSOR_AG(NO_SPONSOR) = N - 1;
          IF (N-1) > 132 THEN GO TO OFF_LNTH;
          END SP_LP;
```

```
                                                                        00004360
     /*  PULL THE PRICE OFF OF LINE 50.  */                             00004370
                                                                        00004380
                                                                        00004390
PRICE_SECT:                                                             00004400
          L_CONTRACT_NO,L_GRANT_NO,L_PA_NO = 0;                         00004410
          CALL SEPERATE (50,N50,5,REP_BUR_SECT,LONG);                   00004420
          N = INDEX(LONG,TB);                                           00004430
          IF N = 0 THEN GO TO MISSCR;                                   00004440
          IF N = 1 THEN GO TO OFF_LNTH;                                 00004450
          PRICE = SUBSTR(LONG,1,N-1);                                   00004460
          L_PRICE = N - 1;                                              00004470
          IF (N-1) > ((N50-49)*132/2) THEN GO TO OFF_LNTH;              00004480
          LONG = SUBSTR(LONG,N+1);                                      00004490
                                                                        00004500
                                                                        00004510
     /*  CHECK FOR A CONTRACT NUMBER.  */                               00004520
                                                                        00004530
                                                                        00004540
          N = INDEX(LONG,TB);                                           00004550
          IF N = 0 THEN GO TO MISSCR;                                   00004560
          IF N = 1 THEN GO TO MISS1;                                    00004570
          CONTRACT_NO = SUBSTR(LONG,1,N-1);                             00004580
          L_CONTRACT_NO = N - 1;                                        00004590
          IF (N-1) > ((N50-49)*132/3) THEN GO TO OFF_LNTH;              00004600
MISS1:    LONG = SUBSTR(LONG,N+1);                                      00004610
                                                                        00004620
                                                                        00004630
     /*  CHECK FOR A GRANT NUMBER.  */                                  00004640
                                                                        00004650
                                                                        00004660
          N = INDEX(LONG,TB);                                           00004670
          IF N = 0 THEN GO TO MISSCR;                                   00004680
          IF N = 1 THEN GO TO MISS2;                                    00004690
          GRANT_NO = SUBSTR(LONG,1,N-1);                                00004700
          L_GRANT_NO = N - 1;                                           00004710
          IF (N-1) > ((N50-49)*132/3) THEN GO TO OFF_LNTH;              00004720
MISS2:    LONG = SUBSTR(LONG,N+1);                                      00004730
                                                                        00004740
                                                                        00004750
     /*  CHECK FOR A PA NUMBER.  */                                     00004760
                                                                        00004770
                                                                        00004780
          N = INDEX(LONG,CR);                                           00004790
          IF N = 0 THEN GO TO MISSCR;                                   00004800
          IF N = 1 THEN GO TO REP_BUR_SECT;                             00004810
          PA_NO = SUBSTR(LONG,1,N-1);                                   00004820
          L_PA_NO = N - 1;                                              00004830
          IF (N-1) ¬= 2 THEN GO TO OFF_LNTH;                            00004840
                                                                        00004850
                                                                        00004860
     /*  CHECK FOR A REPORT NUMBER.  */                                 00004870
                                                                        00004880
                                                                        00004890
REP_BUR_SECT:                                                           00004900
          L_REPORT_NO,L_BUREAU_NO = 0;                                  00004910
          CALL SEPERATE(60,N60,6,AVL_SECT,LONG);                        00004920
          N = INDEX(LONG,TB);                                           00004930
          IF N = 0 THEN GO TO MISSCR;                                   00004940
          IF N = 1 THEN GO TO BUR;                                      00004950
          REPORT_NO = SUBSTR(LONG,1,N-1);                               00004960
          L_REPORT_NO = N - 1;                                          00004970
```

```
          IF (N-1) > ((N60-59)*132/2) THEN GO TO OFF_LNTH;                00004980
BUR:      LONG = SUBSTR(LONG,N+1);                                        00004990
                                                                          00005000
                                                                          00005010
     /*--CHECK FOR A BUREAU NUMBER.  */                                   00005020
                                                                          00005030
                                                                          00005040
          N = INDEX(LONG,CR);                                             00005050
          IF N = 0 THEN GO TO MISSCR;                                     00005060
          IF N = 1 THEN GO TO AVL_SECT;                                   00005070
          BUREAU_NO = SUBSTR(LONG,1,N-1);                                 00005080
          L_BUREAU_NO = N - 1;                                            00005090
          IF (N-1) > ((N60-59)*132/2) THEN GO TO OFF_LNTH;                00005100
                                                                          00005110
                                                                          00005120
     /*  CHECK FOR AN AVAILABILITY STRING. */                             00005130
                                                                          00005140
                                                                          00005150
AVL_SECT:                                                                 00005160
          L_AVAILABILITY = 0;                                             00005170
          CALL SEPERATE (70,N70,7,JC_SECT,LONG);                          00005180
          N = INDEX(LONG,CR);                                             00005190
          IF N = 0 THEN GO TO MISSCR;                                     00005200
          AVAILABILITY = SUBSTR(LONG,1,N-1);                              00005210
          L_AVAILABILITY = N - 1;                                         00005220
                                                                          00005230
                                                                          00005240
     /*  CHECK FOR A JOURNAL CITATION.  */                                00005250
                                                                          00005260
                                                                          00005270
JC_SECT:                                                                  00005280
          L_JOUR_CIT = 0;                                                 00005290
          CALL SEPERATE (80,N80,8,PUB_SECT,LONG);                         00005300
          N = INDEX(LONG,CR);                                             00005310
          IF N = 0 THEN GO TO MISSCR;                                     00005320
          JOUR_CIT = SUBSTR(LONG,1,N-1);                                  00005330
          L_JOUR_CIT = N - 1;                                             00005340
                                                                          00005350
                                                                          00005360
     /*   GET THE PUBLICATION DATE FROM LINE 90.  IT ALWAYS HAS A LENGTH  OF00005370
          7.  */                                                          00005380
                                                                          00005390
                                                                          00005400
PUB_SECT:                                                                 00005410
          L_PUB_DATE,L_PAGE,L_DESC_NOTE = 0;                              00005420
          CALL SEPERATE (90,N90,9,DESC_SECT,LONG);                        00005430
          N = INDEX(LONG,TB);                                             00005440
          IF N = 0 THEN GO TO MISSCR;                                     00005450
          IF N = 1 THEN DO;                                               00005460
                          PUB_DATE = '       ';                           00005470
                          L_PUB_DATE = 7;                                 00005480
                          GO TO GP;                                       00005490
                        END;                                              00005500
          PUB_DATE = SUBSTR(LONG,1,N-1);                                  00005510
          L_PUB_DATE = N - 1;                                             00005520
          IF (N-1) ¬= 7 THEN GO TO OFF_LNTH;                              00005530
GP:       LONG = SUBSTR(LONG,N + 1);                                      00005540
                                                                          00005550
                                                                          00005560
     /*   GET THE PAGE NUMBER FROM LINE 90.  */                           00005570
                                                                          00005580
                                                                          00005590
```

```
          N = INDEX(LONG,T8);                                              00005600
          IF N = 0 THEN GO TO MISSCR;                                      00005610
          IF N = 1 THEN DO;                                                00005620
                        PAGE = '   ';                                      00005630
                        L_PAGE = 3;                                        00005640
                        GO TO DS;                                          00005650
                        END;                                               00005660
          PAGE = SUBSTR(LONG,1,N-1);                                       00005670
          L_PAGE = N - 1;                                                  00005680
          IF (N-1) > 16 THEN GO TO OFF_LNTH;                               00005690
DS:       LONG = SUBSTR(LONG,N+1);                                         00005700
                                                                           00005710
                                                                           00005720
/*   CHECK FOR THE DESCRIPTIVE NOTE.  */                                   00005730
                                                                           00005740
                                                                           00005750
          N = INDEX(LONG,CR);                                              00005760
          IF N = 0 THEN GO TO MISSCR;                                      00005770
          IF N = 1 THEN GO TO DESC_SECT;                                   00005780
          DESC_NOTE=SUBSTR(LONG,1,N-1);                                    00005790
          L_DESC_NOTE = N - 1;                                             00005800
                                                                           00005810
                                                                           00005820
/*   GET THE DESCRIPTOR STRING.  */                                        00005830
                                                                           00005840
                                                                           00005850
DESC_SECT:                                                                 00005860
          L_DESCRIP = 0;                                                   00005870
          IND = 10;                                                        00005880
          CALL SEPERATE (100,N100,10,ID_SECT,DESCRIP);                     00005890
          NO_DES = 0;                                                      00005900
RET2:     N = INDEX(DESCRIP,CR);                                           00005910
          IF N = 0 THEN GO TO MISSCR;                                      00005920
          DESCRIP = SUBSTR(DESCRIP,1,N-1);                                 00005930
          L_DESCRIP = N - 1;                                               00005940
          IF NO_DES = 1 THEN GO TO SEARCH;                                 00005950
                                                                           00005960
                                                                           00005970
/*   CHECK FOR THE EXISTANCE OF IDENTIFIERS.  */                           00005980
                                                                           00005990
                                                                           00006000
ID_SECT:                                                                   00006010
          L_IDENT = 0;                                                     00006020
          IND = 11;                                                        00006030
          IMP = 0;                                                         00006040
          CALL SEPERATE (200,N200,11,SEARCH,LONG);                         00006050
          IMP = 1;                                                         00006060
RETRN:    IF LONG = '' THEN GO TO SEARCH;                                  00006070
          N = INDEX(LONG,CR);                                              00006080
          IF N = 0 THEN GO TO MISSCR;                                      00006090
          IDENT = SUBSTR(LONG,1,N-1);                                      00006100
          L_IDENT = N - 1;                                                 00006110
          GO TO SEARCH;                                                    00006120
                                                                           00006130
                                                                           00006140
/*   ERROR ROUTINE FOR HAVING A MISMATCHED CARD NUMBER ON THE DATA         00006150
     CARD.  */                                                             00006160
                                                                           00006170
                                                                           00006180
ERR:      DO;                                                              00006190
          PUT FILE (ERRE) EDIT                                             00006200
            ('MISMATCHED CARD NUMBER IN DOCUMENT ',VT)                     00006210
```

```
          (SKIP(2),A(35),A(10));                                          00006220
        GO TO SKIP;                                                       00006230
        END ERR;                                                          00006240
                                                                          00006250
------------------------------------------------------------------------- 00006260
  /*   ERROR ROUTINE FOR A MISSING TAB OR END OF LINE CHARACTER ON A DATA00006270
       CARD.  */                                                          00006280
                                                                          00006290
                                                                          00006300
  MISSCR: DO;                                                             00006310
          PUT FILE (ERRE) EDIT                                            00006320
            ('MISSING END OF LINE OR TAB CHARACTER IN ',VT)               00006330
            (SKIP(2),A(37),A(10));                                        00006340
          IF IMP = 2 THEN GO TO SKIP;                                     00006350
          IF IMP=1 THEN READ FILE (CARDS) IGNORE (1);                     00006360
          GO TO START;                                                    00006370
          END;                                                            00006380
                                                                          00006390
                                                                          00006400
  /*   ERROR ROUTINE FOR EXCEEDING A STRING LENGTH OR DIMENSION.       */00006410
                                                                          00006420
                                                                          00006430
  OFF_LNTH: DO;                                                           00006440
            PUT FILE (ERRE) EDIT                                          00006450
              ('LENGTH OF VARIABLE OR DIMENSION IS OFF IN ',VT)           00006460
              (SKIP(2),A(42),A(10));                                      00006470
            GO TO SKIP;                                                   00006480
            END;                                                          00006490
                                                                          00006500
                                                                          00006510
  /*   ERROR ROUTINE FOR MISMATCHED VT NUMBER ON DATA CARDS. IF THE MISMA00006520
       DUE TO A BEGIN DOCUMENT CARD, NO ERROR IS FLAGGED.  */             00006530
                                                                          00006540
                                                                          00006550
  ERROR:  IF SUBSTR(VT_NO,1,1) ¬= BD THEN GO TO BAD;                      00006560
          IF (IND = 10) & (VT_NO ¬= VT) THEN DO;                          00006570
                                             IMP = 0;                     00006580
                                             L_IDENT = 0;                 00006590
                                             NO_DES = 1;                  00006600
                                             GO TO RET2;                  00006610
                                             END;                         00006620
          IF (IND = 11) & (VT_NO ¬= VT) THEN GO TO RETRN;                 00006630
          BAD:   PUT FILE (ERRE) SKIP(2);                                 00006640
                 PUT FILE (ERRE) EDIT                                     00006650
                   ('DOCUMENT NO. DOES NOT MATCH ON ONE CARD IN ',VT)     00006660
                     (SKIP,A(43),A(10));                                  00006670
                 GO TO SKIP;                                              00006680
                                                                          00006690
                                                                          00006700
  /*   ERROR ROUTINE FOR MISSING INFORMATION THAT IS REQUIRED FOR EACH    00006710
       DOCUMENT.  TITLE, PUBLICATION DATE, ETC. ARE SUCH REQUIRED         00006720
       INFORMATION.  */                                                   00006730
------------------------------------------------------------------------- 00006740
                                                                          00006750
  NO_INF: PUT FILE (ERRE) EDIT                                            00006760
            ('EXPECTED INFORMATION MISSING FROM DOCUMENT ',VT)            00006770
            (SKIP(2),A(34),A(10));                                        00006780
                                                                          00006790
------------------------------------------------------------------------- 00006800
                /* THESE STATEMENTS SKIP TO THE BEGINNING OF THE         00006810
                   NEXT DOCUMENT.                                  */     00006820
                                                                          00006830
```

```
SKIP: READ FILE (CARDS) INTO (STR);                                    00006840
      GET STRING (STR) EDIT (BEG)  (A(1));                             00006850
      IF BEG = BD THEN GO TO START;                                    00006860
      GO TO SKIP;                                                      00006870
                                                                       00006880
                                                                       00006890
                                                                       00006900
                                                                       00006910
/*  SEARCH FOR A DOCUMENT'S ED NUMBER AND RESUME PAGE. IF NO CARD IS F00006920
    FOR THE DOCUMENT, AN ERROR IS RAISED.  */                          00006930
                                                                       00006940
                                                                       00006950
SEARCH: DO I = 1 TO K;                                                 00006960
          IF V(I) = VT THEN DO;                                        00006970
                            ED_NO = E(I);                              00006980
                            PAGE_NO = PG(I);                           00006990
                            GO TO PGE;                                 00007000
                            END;                                       00007010
        END SEARCH;                                                    00007020
        PUT FILE (ERRE) EDIT ('NO MATCH MADE ON ED NUMBER FOR ',VT)    00007030
          (A(31),A(10));                                               00007040
        IF IMP=1 THEN READ FILE (CARDS) IGNORE (1);                    00007050
        GO TO START;                                                   00007060
                                                                       00007070
                                                                       00007080
/*  WRITE THE DOCUMENT'S DATA ON MASTER IN ITS OWN RECORD.  LIST THE R00007090
    ON THE PRINTER AS WELL. KEEP A RECORD OF THE NUMBER OF RECORDS ON 00007100
    START PROCESSING OF NEXT DOCUMENT.  */                             00007110
                                                                       00007120
                                                                       00007130
PGE:  L = NO_AUTHOR + NO_CORP_AUTHOR + NO_SPONSOR + 17;                00007140
      PUT STRING (CAT) EDIT (VT,CODE_NO,ED_NO,SEASON,YEAR,PAGE_NO,(    00007150
        AUTHOR(I) DO I = 1 TO NO_AUTHOR),TITLE,(CORP_AUTHOR(I) DO I =00007160
        1 TO NO_CORP_AUTHOR),(SPONSOR_AG(I) DO I = 1 TO NO_SPONSOR),   00007170
        PRICE,CONTRACT_NO,GRANT_NO,PA_NO,REPORT_NO,BUREAU_NO,          00007180
        AVAILABILITY,JOUR_CIT,PUB_DATE,PAGE,DESC_NOTE,DESCRIP,IDENT,   00007190
        SEP,NO_AUTHOR,(L_AUTHOR(I) DO I=1 TO NO_AUTHOR),L_TITLE,       00007200
        NO_CORP_AUTHOR,(L_CORP_AUTHOR(I) DO I=1 TO NO_CORP_AUTHOR),    00007210
        NO_SPONSOR,(L_SPONSOR_AG(I) DO I=1 TO NO_SPONSOR),L_PRICE,     00007220
        L_CONTRACT_NO,L_GRANT_NO,L_PA_NO,L_REPORT_NO,L_BUREAU_NO,      00007230
        L_AVAILABILITY,L_JOUR_CIT,L_PUB_DATE,L_PAGE,L_DESC_NOTE,       00007240
        L_DESCRIP,L_IDENT)                                             00007250
        (A(10),A(17),A(20),A(3),A(4),A(4),(NO_AUTHOR) A,A(L_TITLE),    00007260
        (NO_CORP_AUTHOR) A,(NO_SPONSOR) A,A(L_PRICE),A(L_CONTRACT_NO)00007270
        ,A(L_GRANT_NO),A(L_PA_NO),A(L_REPORT_NO),A(L_BUREAU_NO),       00007280
        A(L_AVAILABILITY),A(L_JOUR_CIT),                               00007290
        A(L_PUB_DATE),A(L_PAGE),A(L_DESC_NOTE),A(L_DESCRIP),           00007300
        A(L_IDENT),A(2),(L) F(4));                                     00007310
      WRITE FILE (MASTER) FROM (CAT);                                  00007320
      PUT FILE (SYSPRINT) SKIP(2);                                     00007330
      PUT FILE (SYSPRINT) EDIT (CAT) (A(LENGTH(CAT)));                 00007340
      NO_RECS = NO_RECS + 1;                                           00007350
      IF IMP=1 THEN READ FILE (CARDS) IGNORE (1);                      00007360
      GO TO START;                                                     00007370
                                                                       00007380
                                                                       00007390
/*  THIS PROCEDURE READS DATA CARDS STARTING AT LINE -K- AND CONTINUES00007400
    THE LINE NUMBER IS UNDER -L-.  IT PUTS THE DATA FROM THE SECTION I00007410
    STRING -BIG-, WHICH IS RETURNED TO THE MAIN PROGRAM TO BE PROCESSE00007420
    TESTED FOR ERRORS.  */                                             00007430
                                                                       00007440
                                                                       00007450
```

```
SEPERATE: PROCEDURE (K,L,IND,LBL_NXT,BIG);                              00007460
        DECLARE                                                         00007470
          K                                                             00007480
              FIXED BINARY (31,0),                                      00007490
          L                                                             00007500
              FIXED BINARY (31,0),                                      00007510
          I                                                             00007520
              FIXED BINARY (31,0),                                      00007530
          J                                                             00007540
              FIXED BINARY (31,0),                                      00007550
          BIG                 CHARACTER (660) VARYING,                  00007560
          LITTLE                                                        00007570
              CHARACTER (66),                                           00007580
          IND                                                           00007590
              FIXED BINARY (31,0),                                      00007600
          LBL_NXT                                                       00007610
              LABEL (CORP_SECT,SPON_SECT,PRICE_SECT,REP_BUR_SECT,       00007620
                     AVL_SECT,JC_SECT,PUB_SECT,DESC_SECT,ID_SECT,       00007630
                     SEARCH,TLE_SECT),                                  00007640
          ;                                                             00007650
        BIG = ' ';                                                      00007660
LOOP:   DO I = K TO L;                                                  00007670
        IF TAB = 1 THEN GO TO LN_CK;                                    00007680
        READ FILE (CARDS) INTO (STR);                                   00007690
        GET STRING (STR) EDIT (VT_NO,LINE_NO,CARD_NO)                   00007700
          (A(10),F(3),F(1));                                            00007710
        IF VT_NO ¬= VT THEN GO TO ERROR;                                00007720
LN_CK:  IF LINE_NO = I THEN GO TO CD_CK;                                00007730
        IF (IND=2)|(IND=9)|(IND=10)|(IND=5)                             00007740
                                    THEN DO;                            00007750
                                         IF I = K THEN GO TO NO_INF;    00007760
                                         TAB = 1;                       00007770
                                         RETURN;                        00007780
                                         END;                           00007790
                                    ELSE DO;                            00007800
                                         TAB = 1;                       00007810
                                         IF I = K THEN GO TO LBL_NXT;   00007820
                                         RETURN;                        00007830
                                         END;                           00007840
CD_CK:  DO J = 1 TO 2;                                                  00007850
        IF J = 1 THEN GO TO NO_RD;                                      00007860
        READ FILE (CARDS) INTO (STR);                                   00007870
        GET STRING (STR) EDIT (VT_NO,LINE_NO,CARD_NO)                   00007880
          (A(10),F(3),F(1));                                            00007890
        IF VT_NO ¬= VT THEN GO TO ERROR;                                00007900
        IF LINE_NO ¬= I THEN DO;                                        00007910
                             TAB = 1;                                   00007920
                             RETURN;                                    00007930
                             END;                                       00007940
NO_RD:  IF CARD_NO ¬= J THEN GO TO ERR;                                 00007950
        GET STRING (STR) EDIT (LITTLE) (X(14),A(66));                   00007960
        BIG = BIG || LITTLE;                                            00007970
        END CD_CK;                                                      00007980
        TAB = 0;                                                        00007990
END_LP: END LOOP;                                                       00008000
        RETURN;                                                         00008010
        END SEPERATE;                                                   00008020
                                                                        00008030
                                                                        00008040
/*   PRINT OUT THE NUMBER OF RECORDS ON MASTER. CLOSE ALL FILES.  */    00008050
                                                                        00008060
                                                                        00008070
```

```
STOP:   PUT DATA (NO_RECS);                                              00008080
        IF NO_RECS ¬= K THEN PUT FILE (ERRE) EDIT                        00008090
          ('NOT ALL DOCUMENTS WERE PROCESSED. THERE SHOULD BE ',K)       00008100
          (SKIP(2),A(50),F(4));                                          00008110
        CLOSE FILE (MASTER), FILE (ERRE);                                00008120
        CLOSE FILE (CARDS);                                              00008130
        END WORK;                                                        00008140
        END BEGIN;                                                       00008150
```

ABSTRACT

```
        /*                                                                  00000010
              PURPOSE:  READ ABSTRACT DATA CARDS AND CREATE AN ABSTRACT R00000020
                        FOR EACH DOCUMENT.  RECORD CONSISTS OF CLEARINGH000000030
                        ACCESSION NUMBER, 2 DIGIT CODE TO BE USED FOR SOR00000040
                        AND THE ABSTRACT.                                 00000050
        */                                                                00000060
   GENABS: PROCEDURE OPTIONS(MAIN);                                       00000070
   DCL LAST CHAR(4),                                                      00000080
            CODE CHAR(2),                                                 00000090
       A_OR_BLNK CHAR(1),                                                 00000100
             1 RESUME_CARD,                                               00000110
               2 VT CHAR(10),                                             00000120
               2 CARD_ID CHAR(4) INITIAL('    '),                         00000130
               2 DATA CHAR(66),                                           00000140
              VT_CURRENT CHAR(10),                                        00000150
             RESUME CHAR(2000) VAR;                                       00000160
   ON ENDFILE(CARDS) GO TO SORT;                                          00000170
   GET FILE(SYSIN) EDIT(A_OR_BLNK,CODE)(A(1),X(78),A(1));                 00000180
       /* * * * * * * * * * * */                                         00000190
       /* * * * * * * * * * * */                                         00000200
       /*                     */                                         00000210
       /* LOOP TO READ RESUME */                                         00000220
       /*     AND FORM RECORD */                                         00000230
       /*                     */                                         00000240
       /* * * * * * * * * * * */                                         00000250
     FIRST: READ FILE(CARDS) INTO(RESUME_CARD);                         00000260
    IF SUBSTR(VT,3,1)¬=' '|SUBSTR(VT,7,1)¬=' ' THEN DO;                 00000270
     PUT FILE(SYSPRINT) SKIP(2);                                        00000280
    BADVT1: PUT FILE(SYSPRINT) EDIT(VT,CARD_ID,DATA)(SKIP(1),A(10),A(4),00000290
              A(66));                                                   00000300
            READ FILE(CARDS) INTO(RESUME_CARD);                         00000310
            IF SUBSTR(VT,1,1)='¬' THEN GO TO FIRST;                     00000320
            GO TO BADVT1;                                               00000330
                                                  END;                  00000340
   IF SUBSTR(VT,1,1)='¬' THEN GO TO FIRST;                              00000350
            IF A_OR_BLNK='A' THEN I=5; ELSE I=8;                        00000360
            GET STRING(DATA) EDIT(CODE)(X(I),A(2));                     00000370
            READ FILE(CARDS) INTO(RESUME_CARD);                         00000380
            RESUME=VT||CODE||DATA;                                      00000390
            VT_CURRENT=VT;                                              00000400
            I=INDEX(DATA,'|');                                          00000410
            LAST=CARD_ID;                                               00000420
            IF I=0 THEN GO TO AGAIN;                                    00000430
            RESUME=SUBSTR(RESUME,1,I+12);                               00000440
            WRITE FILE(SORTIN) FROM(RESUME);                            00000450
            GO TO FIRST;                                                00000460
       /*  READ ONE CARD OF ABSTRACT CARD DECK    */                   00000470
     AGAIN: READ FILE(CARDS) INTO(RESUME_CARD);                         00000480
       /*  CHECK FOR ERROR IN CLEARINGHOUSE ACCESSION NUMBER    */       00000490
    IF SUBSTR(VT,3,1)¬=' '|SUBSTR(VT,7,1)¬=' ' THEN DO;                 00000500
     PUT FILE(SYSPRINT) SKIP(2);                                        00000510
     BADVT2: PUT FILE(SYSPRINT) EDIT(VT,CARD_ID,DATA)(SKIP(1),A(10),A(4),00000520
              A(66));                                                   00000530
            READ FILE(CARDS) INTO(RESUME_CARD);                         00000540
            IF SUBSTR(VT,1,1)='¬' THEN GO TO FIRST;                     00000550
            GO TO BADVT2;                                               00000560
                                                  END;                  00000570
         /*   CHECK FOR DOCUMENT SEPERATOR  */                          00000580
   IF SUBSTR(VT,1,1)='¬' THEN GO TO FIRST;                              00000590
            I=INDEX(DATA,'|');                                          00000600
         /*  CHECK FOR END OF ABSTRACT */                               00000610
          IF VT¬=VT_CURRENT|LAST>=CARD_ID THEN DO;                      00000620
```

```
      /*    CHECK CARD FOR ERROR IN CLEARINGHOUSE ACCESSION NUMBER OR SEQU00000630
            */                                                          00000640
           PUT FILE(SYSPRINT) EDIT(LAST)(SKIP(2),A);                    00000650
    BADID: PUT FILE(SYSPRINT) EDIT(RESUME_CARD)(SKIP,A(10),A(4),A(66)); 00000660
           READ FILE(CARDS) INTO(RESUME_CARD);                          00000670
           IF SUBSTR(VT,1,1)='¬' THEN GO TO FIRST;                      00000680
           GO TO BADID;                                                 00000690
           END;                                                         00000700
            IF I=0 THEN RESUME=RESUME||DATA:                            00000710
                   ELSE DO;                                             00000720
                        RESUME=RESUME||SUBSTR(DATA,1,I);                00000730
                        WRITE FILE(SORTIN) FROM(RESUME);                00000740
                        GOTO FIRST;                                     00000750
                      END;                                              00000760
      LAST=CARD-ID;                                                     00000770
           GO TO AGAIN;                                                 00000780
         /* * * * * * * * * * * * */                                    00000790
         /* * * * * * * * * * * * */                                    00000800
    SORT: END;                                                          00000810
```

RESUME PRINT

```
/*                                                                      00000010
                                                                        00000020
        THIS SECTION CREATES THE RESUME PRINTOUT                        00000030
                                                                        00000040
*/                                                                      00000050
                                                                        00000060
                                                                        00000070
SECT1: PROCEDURE OPTIONS(MAIN);                                         00000080
                                                                        00000090
DCL A_OR_BLNK CHAR(1);                                                  00000100
                                                                        00000110
 /*                                                                     00000120
       READ CONTROL CARDS                                               00000130
 */                                                                     00000140
                                                                        00000150
GET FILE(SYSIN) DATA(A_OR_BLNK);                                        00000160
 /*                                                                     00000170
       START PROCESSING RECORDS FOR PRINTED SECTION                     00000180
 */                                                                     00000190
DCL BUF_VT CHAR(10),                                                    00000200
    CODE CHAR(17),                                                      00000210
         AIM_CODE CHAR(2) DEFINED CODE POSITION(6),                     00000220
         ARM_CODE CHAR(2) DEFINED CODE POSITION(9),                     00000230
      BUF_ED CHAR(20),                                                  00000240
      BUF_PUB CHAR(1),                                                  00000250
      BUF_SEASON CHAR(6),                                               00000260
      RESUME_PG_NO FIXED BINARY(31,0),                                  00000270
      CARD CHAR(80),                                                    00000280
      BUF2 CHAR(3678) VAR,                                              00000290
      IN_BUFFER CHAR(3736) VAR,                                         00000300
            BRAKUP                                                      00000310
              ENTRY(CHAR(2000) VAR),                                    00000320
     (VT_ED_SKIP,BREAK,I,J,#_P_AUTHORS,LEN_P_AUTHOR(2),TITLE_LEN,#SA,   00000330
      #_C_AUTHORS,LEN_C_AUTHOR(5),LEN_SA(3),LEN_PRICE,LEN_CONTRACT,     00000340
      LEN_JC,LEN_DATE,LEN_PAGES,LEN_DESC_NOTE,LEN_DESC,LEN_IDENT,       00000350
      PAGE_WIDTH) FIXED BINARY(31,0),                                   00000360
       (SPACE_LINES,INDENT) FIXED BINARY(31,0),                         00000370
      (FIRST_LINE,TOP_LINE,CON_1ST_LINE) FIXED BINARY(31,0),            00000380
      RESUME CHAR(2000) VAR,                                            00000390
      (VT_COL,PRICE_COL,CONTRACT_COL,GRANT_COL,PA_COL) FIXED BINARY(15), 00000400
      TITLE CHAR(396) VAR,                                              00000410
      (REPORT_COL,BUREAU_COL,LEN_HEADING) FIXED BINARY(15,0),           00000420
      INDEX_PAGE FIXED BINARY(31,0),                                    00000430
      INDEX_PAGE_CHACTER CHAR(4),                                       00000440
      BLANKS CHAR(132) INITIAL( (132)' ' ),                             00000450
     (LINE_50, LINE_60, LINE_90) CHAR(198) VAR,                         00000460
      CURRENT_CODE CHAR(2) INITIAL('  '),                               00000470
      OLD_CODE CHAR(2),                                                 00000480
      HEADING CHAR(20) VAR,                                             00000490
      (TOTAL_LINES,TOTAL_COLUMNS) FIXED BINARY(31,0), /* OUTPUT DIMEN.*/ 00000500
      UNDER CHAR(10) INITIAL( (10) '_'),                                00000510
       INITIALS CHAR(4) INITIAL('    '),                                00000520
      TOP CHAR(4);                                                      00000530
      DCL TEMP_STRING CHAR(2000) VAR;                                   00000540
      DCL VT_OUT(20) CHAR(10),                                          00000550
      RES_CODE CHAR(2),                                                 00000560
       IVT FIXED BINARY(15,0);                                          00000570
                                                                        00000580
 /*                                                                     00000590
       EOF CONTROL                                                      00000600
 */                                                                     00000610
                                                                        00000620
```

```
ON ENDFILE(MASTER)  BEGIN;                                              00000630
                    SIGNAL ENDPAGE(OUT);                                00000640
                    GOTO FINISH;                                        00000650
                    END;                                                00000660
ON ENDFILE(SORTOUT) BEGIN;                                              00000670
                    PUT FILE(SYSPRINT) LIST('RESUME HAS FEWER RECORDS');00000680
                    GO TO FINISH;                                       00000690
                    END;                                                00000700
ON ENDPAGE(OUT) BEGIN;                                                  00000710
      CURRENT_CODE=OLD_CODE;                                            00000720
      IF TOP='LEFT' THEN DO;                                            00000730
                         TOP='RITE';                                    00000740
                         J1=INDENT;                                     00000750
                         J2=LEN_HEADING+INDENT;                         00000760
                         END;                                           00000770
                    ELSE DO;                                            00000780
                         TOP='LEFT';                                    00000790
                         J1=LEN_HEADING+INDENT+10;                      00000800
                         J2=INDENT;                                     00000810
                         END;                                           00000820
      PUT FILE(OUT) PAGE LINE(TOP_LINE) EDIT(INDEX_PAGE,HEADING)        00000830
            (COLUMN(J1),F(4),SKIP(0),COLUMN(J2),A);                     00000840
      PUT FILE(OUT) SKIP(2);                                            00000850
                                   TOP_LINE=CON_1ST_LINE;               00000860
      INDEX_PAGE=INDEX_PAGE+1;                                          00000870
                END;                                                    00000880
      GET FILE(SYSIN) DATA COPY;                                        00000890
    GET FILE(BADVT) EDIT(IVT,(VT_OUT(J) DO J=1 TO IVT))(F(4),20 A(10));00000900
      KEY=0;                                                            00000910
  OPEN FILE(OUT) PRINT PAGESIZE(TOTAL_LINES);                           00000920
    LEN_HEADING=PAGE_WIDTH-LENGTH(HEADING);                             00000930
                                   TOP_LINE=FIRST_LINE;                 00000940
                                                                        00000950
                                                                        00000960
   /*                                                                   00000970
      READ BIBLIOGRAPHIC RECORD                                         00000980
   */                                                                   00000990
                                                                        00001000
  LOOP: READ FILE(MASTER) INTO(IN_BUFFER);                              00001010
      GET STRING(IN_BUFFER) EDIT(BUF_VT,CODE,BUF_ED,BUF_PUB,BUF_SEASON,00001020
        RESUME_PG_NO,BUF2)(A(10),A(17),A(20),A(1),A(6),F(4),            00001030
        A(LENGTH(IN_BUFFER)-58));                                       00001040
                                                                        00001050
   /*                                                                   00001060
        CHECK IF CATEGORY OR FIELD HAS CHANGED                          00001070
              IF SO,                                                    00001080
                   PUT LAST PAGE ON PRINT FILE                          00001090
                   AND SET UP NEW PAGE                                  00001100
                   THEN READ ABSTRACT RECORD                            00001110
              IF NOT, READ ABSTRACT RECORD                              00001120
   */                                                                   00001130
    IF BUF_PUB='A'&AIM_CODE¬=CURRENT_CODE THEN DO;                      00001140
                   IF TOP='LEFT' THEN TOP='RITE';                       00001150
                                 ELSE TOP='LEFT';                       00001160
                   INDEX_PAGE=INDEX_PAGE+1;                             00001170
                   OLD_CODE=AIM_CODE;                                   00001180
                   SIGNAL ENDPAGE(OUT);                                 00001190
                   GO TO READ_RESUME;   END;                            00001200
      IF BUF_PUB¬='A'&ARM_CODE¬=CURRENT_CODE THEN DO;                   00001210
              IF TOP='LEFT'  THEN TOP='RITE';                           00001220
                             ELSE TOP='LEFT';                           00001230
                   INDEX_PAGE=INDEX_PAGE+1;                             00001240
```

```
                                    OLD_CODE=ARM_CODE;                          00001250
                                    SIGNAL ENDPAGE(OUT);                        00001260
                                  END;                                          00001270
                                                                                00001280
        IF KEY=1 THEN GO TO NO_RES;                                             00001290
                                                                                00001300
    /*                                                                          00001310
            READ ABSTRACT RECORD                                                00001320
    */                                                                          00001330
                                                                                00001340
     READ_RESUME:                                                               00001350
       READ FILE(SORTOUT) INTO(RESUME);                                         00001360
   NO_RES: KEY=0;                                                               00001370
       RES_CODE=SUBSTR(RESUME,11,2);                                            00001380
       IF RES_CODE<'00' THEN GO TO READ_RESUME;                                 00001390
     DO I=1 TO IVT;                                                             00001400
       IF BUF_VT=VT_OUT(I) THEN GO TO LOOP;                                     00001410
     END;                                                                       00001420
       IF SUBSTR(RESUME,1,10)>BUF_VT&(CURRENT_CODE=ARM_CODE|CURRENT_CODE=       00001430
          ARM_CODE) THEN DO;                                                    00001440
           PUT FILE(SYSPRINT) LIST('**ERROR ON MATCH ',BUF_VT,RESUME);          00001450
       IF RES_CODE¬=CURRENT_CODE THEN DO; KEY=1;                                00001460
                                      GO TO LOOP;                               00001470
                                  END;                                          00001480
                END;                                                            00001490
                                                                                00001500
    /*                                                                          00001510
          CHECK IF ABSTRACT VT=BIBLIOGRAPHIC VT                                 00001520
                IF YES, CONTINUE                                                00001530
                IF NOT, RETURN TO BIBLIOGRAPHIC RECORD READ-IN                  00001540
    */                                                                          00001550
                                                                                00001560
       IF SUBSTR(RESUME,1,10)¬=BUF_VT THEN DO;                                  00001570
     RESUME=SUBSTR(RESUME,1,10)||BUF_VT;                                        00001580
    PUT FILE(SYSPRINT) LIST('** ERRORS IN VTS RESUME BIB **',RESUME);           00001590
       GO TO READ_RESUME;                                                       00001600
                                  END;                                          00001610
       RESUME=SUBSTR(RESUME,13);                                                00001620
       I=LINENO(OUT);                                                           00001630
   IF I>TOTAL_LINES-4 THEN DO;                                                  00001640
                  SIGNAL ENDPAGE(OUT);                                          00001650
                  I=LINENO(OUT);                                                00001660
                                  END;                                          00001670
       INDEX_PAGE=INDEX_PAGE-1;                                                 00001680
       CARD=' ';                                                                00001690
                                                                                00001700
    /*                                                                          00001710
          PUT RESUME PAGE INTO BIB RECORD AND WRITE RECORD                      00001720
    */                                                                          00001730
     PUT STRING(CARD) EDIT(BUF_VT,BUF_ED,INDEX_PAGE)(A(10),A(20),F(4));         00001740
   SUBSTR(IN_BUFFER,55,4)=SUBSTR(CARD,31,4);                                    00001750
   WRITE FILE(NEWMAS) FROM(IN_BUFFER);                                          00001760
                                                                                00001770
    /*                                                                          00001780
       PUNCH VT-ED RESUME PAGE CARD                                             00001790
    */                                                                          00001800
                                                                                00001810
       WRITE FILE(PUNCH) FROM(CARD);                                            00001820
       INDEX_PAGE=INDEX_PAGE+1;                                                 00001830
       IF I>TOP_LINE+3 THEN I=3; ELSE I=2;                                      00001840
                                                                                00001850
    /*                                                                          00001860
```

```
          PUT VT-ED LINE INTO RESUME RECORD                                    00001870
    */                                                                         00001880
                                                                               00001890
   PUT FILE(OUT) SKIP(I) EDIT(BUF_VT,UNDER,BUF_ED)(COLUMN(VT_COL),A,SKIP(000001900
   ),COLUMN(VT_COL),A,X(VT_ED_SKIP),A);                                        00001910
       I=INDEX(BUF2,'||')+1;                                                   00001920
          LEN_P_AUTHOR=0; LEN_C_AUTHOR=0; LEN_SA=0;                            00001930
          GET STRING(BUF2) EDIT(#_P_AUTHORS,(LEN_P_AUTHOR(J) DO J=1 TO         00001940
                                #_P_AUTHORS WHILE(#_P_AUTHORS>0)),             00001950
                                TITLE_LEN,                                     00001960
                                #_C_AUTHORS,(LEN_C_AUTHOR(J) DO J=1 TO       00001970
                                #_C_AUTHORS WHILE(#_C_AUTHORS>0)),           00001980
                               #SA,(LEN_SA(J) DO J=1 TO #SA WHILE(#SA>0)),   00001990
                                LEN_PRICE,LEN_CONTRACT,LEN_GRANT,LEN_PA,     00002000
                                LEN_REPORT,LEN_BUREAU,                       00002010
                                LEN_AVAIL,                                   00002020
                                LEN_JC,                                      00002030
                                LEN_DATE,LEN_PAGES,LEN_DESC_NOTE,            00002040
                                LEN_DESC,                                    00002050
                                LEN_IDENT)                                   00002060
                               (X(I),28 F(4));                               00002070
          I=1;                                                               00002080
                                                                             00002090
    /*                                                                       00002100
                SET UP PERSONAL AUTHOR LINE                                  00002110
    */                                                                       00002120
                                                                             00002130
        IF #_P_AUTHORS>0 THEN DO;                                            00002140
             IF #_P_AUTHORS=1 THEN CALL BRAKUP(SUBSTR(BUF2,1,                00002150
             LEN_P_AUTHOR(1))||' %@!');                                      00002160
                              ELSE CALL BRAKUP(SUBSTR(BUF2,1,                00002170
                 LEN_P_AUTHOR(1))||'; '||SUBSTR(BUF2,LEN_P_AUTHOR(1)+        00002180
             1,LEN_P_AUTHOR(2))||' %@!');                                    00002190
             I=I+SUM(LEN_P_AUTHOR);                                          00002200
                              END;                                           00002210
          TITLE=SUBSTR(BUF2,I,TITLE_LEN);                                    00002220
          J=INDEX(TITLE,'<');                                                00002230
                                                                             00002240
    /*                                                                       00002250
                SET UP TITLE LINE                                            00002260
    */                                                                       00002270
                                                                             00002280
  TWO_PART_TITLE: IF J=0 THEN GO TO TITLE_PROCESS;                           00002290
                  IF J=1 THEN TITLE=SUBSTR(TITLE,2);                         00002300
                       ELSE TITLE=SUBSTR(TITLE,1,J-1)||SUBSTR(TITLE,J+1      00002310
                         );                                                  00002320
          J=INDEX(TITLE,'>');                                                00002330
          IF J=0 THEN DO;                                                    00002340
             PUT FILE(SYSPRINT) LIST('TITLE ERROR',IN_BUFFER);               00002350
             GO TO LOOP;                                                     00002360
                END;                                                         00002370
          SUBSTR(TITLE,J)=SUBSTR(TITLE,J+2);                                 00002380
          J=INDEX(TITLE,'<');                                                00002390
           GO TO TWO_PART_TITLE;                                             00002400
  TITLE_PROCESS: CALL BRAKUP(TITLE||' %@!');                                 00002410
          PUT FILE(OUT) SKIP(SPACE_LINES);                                   00002420
          I=I+TITLE_LEN;                                                     00002430
                                                                             00002440
    /*                                                                       00002450
                SET UP CORPORATE AUTHOR LINE                                 00002460
    */                                                                       00002470
                                                                             00002480
```

```
      IF #_C_AUTHORS>0 THEN DO J=1 TO #_C_AUTHORS;                          00002490
            CALL BRAKUP(SUBSTR(BUF2,I,LEN_C_AUTHOR(J))||' %@:');            00002500
            I=I+LEN_C_AUTHOR(J);                                            00002510
                                 END;                                       00002520
      IF #SA>0 THEN DO J=1 TO #SA;                                          00002530
            CALL BRAKUP(SUBSTR(BUF2,I,LEN_SA(J))||' %@:');                  00002540
            I=I+LEN_SA(J);                                                  00002550
                  END;                                                      00002560
                                                                            00002570
/*                                                                          00002580
            SET UP PRICE LINE                                               00002590
*/                                                                          00002600
                                                                            00002610
   BEGIN;                                                                   00002620
     IF LEN_PRICE>0 THEN DO;                                                00002630
   LINE_50=SUBSTR(BLANKS,1,PRICE_COL-1);                                    00002640
   LINE_50=LINE_50||SUBSTR(BUF2,I,LEN_PRICE)||SUBSTR(BLANKS,1,              00002650
      CONTRACT_COL);                                                        00002660
            I=I+LEN_PRICE;                                                  00002670
   IF LEN_CONTRACT=0 THEN LINE_50=LINE_50||SUBSTR(BUF2,I,LEN_GRANT);        00002680
                     ELSE LINE_50=LINE_50||SUBSTR(BUF2,I,LEN_CONTRACT)      00002690
                     ;                                                      00002700
            I=I+LEN_CONTRACT;                                               00002710
            I=I+LEN_GRANT;                                                  00002720
   IF LEN_PA¬=0 THEN LINE_50=LINE_50||SUBSTR(BLANKS,1,PA_COL)||             00002730
      SUBSTR(BUF2,I,LEN_PA);                                                00002740
            I=I+LEN_PA;                                                     00002750
   PUT FILE(OUT) EDIT(LINE_50)(SKIP,A);                                     00002760
                              END;                                          00002770
   END;                                                                     00002780
                                                                            00002790
/*                                                                          00002800
            SET UP REPORT AND BUREAU NUMBER LINES                           00002810
*/                                                                          00002820
                                                                            00002830
   BEGIN;                                                                   00002840
     IF LEN_REPORT>0|LEN_BUREAU>0 THEN DO;                                  00002850
   LINE_60=SUBSTR(BLANKS,1,PRICE_COL-1);                                    00002860
IF LEN_REPORT¬=0 THEN                                                       00002870
   LINE_60=LINE_60||SUBSTR(BUF2,I,LEN_REPORT)||SUBSTR(BLANKS,1,             00002880
      LEN_PRICE-LEN_REPORT);                                                00002890
                ELSE                                                        00002900
   LINE_60=LINE_60||SUBSTR(BLANKS,1,LEN_PRICE);                             00002910
      I=I+LEN_REPORT;                                                       00002920
   IF LEN_BUREAU¬=0 THEN                                                    00002930
LINE_60=LINE_60||SUBSTR(BLANKS,1,GRANT_COL)||SUBSTR(BUF2,I,                 00002940
      LEN_BUREAU);                                                          00002950
      I=I+LEN_BUREAU;                                                       00002960
   PUT FILE(OUT) EDIT(LINE_60)(SKIP,A);                                     00002970
                                       END;                                 00002980
   END;                                                                     00002990
                                                                            00003000
/*                                                                          00003010
            SET UP AVAILABILITY LINE                                        00003020
*/                                                                          00003030
                                                                            00003040
   IF LEN_AVAIL>0 THEN CALL BRAKUP(SUBSTR(BUF2,I,LEN_AVAIL)||' %@:');       00003050
      I=I+LEN_AVAIL;                                                        00003060
   IF LEN_JC>0 THEN CALL BRAKUP(SUBSTR(BUF2,I,LEN_JC)||' %@:');             00003070
      I=I+LEN_JC;                                                           00003080
   BEGIN;                                                                   00003090
     PUT STRING(LINE_90) EDIT('PUB DATE - ',SUBSTR(BUF2,I,LEN_DATE),        00003100
```

```
                '   ',SUBSTR(BUF2,I+LEN_DATE,LEN_PAGES))(A(11),A,A(2),A);   00003110
                IF SUBSTR(LINE_90,LEN_DATE+9,2)='   ' THEN SUBSTR(LINE_90,  00003120
                    LEN_DATE+9,2)='ND';                                      00003130
                IF LEN_DESC_NOTE>0 THEN LINE_90=LINE_90||' '||SUBSTR(BUF2   00003140
                    ,I+LEN_DATE+LEN_PAGES,LEN_DESC_NOTE);                    00003150
                I=I+LEN_DATE+LEN_PAGES+LEN_DESC_NOTE;                        00003160
                 CALL BRAKUP(LINE_90||' %@!');                               00003170
          PUT FILE(OUT) SKIP(SPACE_LINES);                                   00003180
      END;                                                                   00003190
                                                                             00003200
     /*                                                                      00003210
                    SET UP DESCRIPTOR AND IDENTIFIER LINES                   00003220
     */                                                                      00003230
                                                                             00003240
      IF LEN_DESC>0 THEN DO;                                                 00003250
           TEMP_STRING='DESCRIPTORS - '||SUBSTR(BUF2,I,LEN_DESC);            00003260
                TEMP_STRING=TEMP_STRING||' %@!';                             00003270
           CALL BRAKUP(TEMP_STRING);                                         00003280
               I=I+LEN_DESC;                                                 00003290
                    END;                                                     00003300
      IF LEN_IDENT>0 THEN DO;                                                00003310
           TEMP_STRING='IDENTIFIERS - '||SUBSTR(BUF2,I,LEN_IDENT);           00003320
                TEMP_STRING=TEMP_STRING||' %@!';                             00003330
           CALL BRAKUP(TEMP_STRING);                                         00003340
                    END;                                                     00003350
                                                                             00003360
     /*                                                                      00003370
                    SET UP ABSTRACT LINE                                     00003380
     */                                                                      00003390
                                                                             00003400
         I=INDEX(RESUME,'|');                                                00003410
        IF I=0 THEN I=LENGTH(RESUME)+1;                                      00003420
         INITIALS=SUBSTR(RESUME,I-5,4);                                      00003430
         RESUME=SUBSTR(RESUME,1,I-1);                                        00003440
              RESUME=RESUME||' %@!';                                         00003450
         PUT FILE(OUT) SKIP(SPACE_LINES);                                    00003460
          TEMP_STRING='ABSTRACT - '||RESUME;                                 00003470
          CALL BRAKUP(TEMP_STRING);                                          00003480
         INITIALS='    ';                                                    00003490
         GO TO LOOP; /* START NEXT DOCUMENT*/                               00003500
FINISH: BEGIN;                                                               00003510
     CLOSE FILE(OUT), FILE(SORTOUT), FILE(MASTER);                           00003520
      PUT FILE(SYSPRINT) DATA(INDEX_PAGE);                                   00003530
END FINISH;                                                                  00003540
                                                                             00003550
                                                                             00003560
     /*                                                                      00003570
            BRAKUP IS USED BY THE MAIN PROGRAM IN BREAKING UP                00003580
                   THE BIBLIOGRAPHIC RECORD FOR THE RESUME.                  00003590
                   RESUME RECORDS ARE PRINTED AND AT A PAGE'S                00003600
                   END THE TOP OF THE NEXT PAGE IS SET UP.                   00003610
     */                                                                      00003620
                                                                             00003630
     BRAKUP: PROCEDURE(LINE);                                                00003640
     DCL LINE CHAR(2000) VAR,                                                00003650
         OUT_LINE CHAR(132) VAR,                                             00003660
     CHECK_INDEX FIXED BINARY(31,0),                                         00003670
         (J,I,K) FIXED BINARY(15,0);                                         00003680
         J=PAGE_WIDTH;                                                       00003690
         OUT_LINE=LINE;                                                      00003700
          IF LENGTH(LINE)<PAGE_WIDTH THEN DO;                                00003710
          I=INDEX(OUT_LINE,' %@!');                                          00003720
```

```
            OUT_LINE=SUBSTR(OUT_LINE,1,I);                                  00003730
          PUT FILE(OUT) EDIT(OUT_LINE)(SKIP,X(INDENT),A);                   00003740
            RETURN;                                                         00003750
                                        END;                                00003760
KLOOP: DO K=1 TO 50;                                                        00003770
       IF J-PAGE_WIDTH>=LENGTH(LINE) THEN RETURN;                           00003780
  ILOOP: DO I=0 TO PAGE_WIDTH-2;                                            00003790
            IF SUBSTR(LINE,J-I,1)=' '|SUBSTR(LINE,J-I,1)='-' THEN DO;       00003800
            IF LENGTH(LINE)-J>=0 THEN                                       00003810
            OUT_LINE=SUBSTR(LINE,J-PAGE_WIDTH+1,PAGE_WIDTH-I);              00003820
                                   ELSE                                     00003830
            OUT_LINE=SUBSTR(LINE,J-PAGE_WIDTH+1);                           00003840
          GO TO NEXT_LINE;                                                  00003850
                                        END;                                00003860
                    END ILOOP;                                              00003870
  NEXT_LINE: J=J+PAGE_WIDTH-I;                                              00003880
        CHECK_INDEX=INDEX(OUT_LINE,'%@!');                                  00003890
       IF CHECK_INDEX>0 THEN DO;                                            00003900
          IF CHECK_INDEX<3 THEN RETURN;                                     00003910
                           ELSE                                             00003920
            OUT_LINE=SUBSTR(OUT_LINE,1,CHECK_INDEX-1);                      00003930
           PUT FILE(OUT) EDIT(OUT_LINE)(SKIP,X(INDENT),A);                  00003940
            RETURN;                                                         00003950
                             END;                                           00003960
          PUT FILE(OUT) EDIT(OUT_LINE)(SKIP,X(INDENT),A);                   00003970
  END KLOOP;                                                                00003980
  END BRAKUP;                                                               00003990
                                                                            00004000
                                                                            00004010
                                                                            00004020
                    END SECT1;                                              00004030
```

MASTER EDIT

```
/*************************************************************************/00000010
/*                                                                       */00000020
/*       E D I T   P R O G R A M   F O R   M A S T E R                   */00000030
/*                                                                       */00000040
/*                                                                       */00000050
/*  PURPOSE:  THE PROGRAM TAKES A MASTER TAPE DATA SET AND COPIES        */00000060
/*            IT ONTO ANOTHER TAPE WHILE EDITING IT.                     */00000070
/*                                                                       */00000080
/*  PROGRAM EXPLANATION:                                                 */00000090
/*    THE PROGRAM IS GIVEN A LIST OF DOCUMENTS WHICH HAVE ERRORS         */00000100
/*    IN THEIR RECORDS ON MASTER.  THE PROGRAM COPIES THE DATA SET       */00000110
/*    ONTO ANOTHER TAPE BUT THE BAD DOCUMENT RECORDS ARE LEFT OFF OF     */00000120
/*    THE NEW TAPE.  WHEN THE COPYING IS COMPLETE, THE BAD DOCUMENTS'    */00000130
/*    DATA CARDS ARE READ IN.  OF COURSE, THE ERRORS ON THESE CARDS      */00000140
/*    HAVE BEEN CORRECTED.  A PROCEDUR  ALMOST IDENTICAL TO THE          */00000150
/*    MASTER PROGRAM IS THEN CALLED TO PROCESS THESE DATA CARDS AND      */00000160
/*    PUT THE DATA ON THE NEW TAPE.  SO THESE DOCUMENTS WHICH HAD        */00000170
/*    ERRORS BEFORE ARE NOW CORRECT AND ARE AT THE END OF THE NEW        */00000180
/*    DATA SET.                                                          */00000190
/*                                                                       */00000200
/*  INPUT TO PROGRAM:                                                    */00000210
/*    -SYSIN-, INPUT ON CARDS                                            */00000220
/*      CARD 1 -- THE NUMBER OF DOCUMENTS WITH BAD RECORDS ON THE        */00000230
/*                MASTER TAPE DATA SET.  COLS 1-4, RIGHT JUSTIFIED.      */00000240
/*                -NA- IS THIS NUMBER.                                   */00000250
/*      CARDS 2 - (NA+1) -- THE VT NUMBERS OF THE DOCUMENTS, ONE TO      */00000260
/*                          A CARD IN COLS 1-5.                          */00000270
/*      CARD NA+2 -- VALUES FOR THE VARIABLES NA, N10, NT, NC, NS,       */00000280
/*                   N50, N60, N70, N80, N90, N100, N200 ARE ON THIS     */00000290
/*                   CARD.  EACH IS RIGHT JUSTIFIED IN CONSECUTIVE       */00000300
/*                   3 COLUMN FIELD WIDTHS.  SEE THE MASTER PROGRAM      */00000310
/*                   DOCUMENTATION FOR THEIR VALUES.                     */00000320
/*      CARD NA+3 -- THE EXPECTED NUMBER OF DOCUMENTS TO BE PROCESSED    */00000330
/*                   THIS RUN.  COLS. 1-5, RIGHT JUSTIFIED.              */00000340
/*      CARD NA+4 -- COLUMN 1 HAS 'A' FOR AIM OR ' ' FOR ARM.            */00000350
/*                   COLUMNS 2-3 HAS A SEASON, 'F ','W ','SP', OR        */00000360
/*                     'SU'.                                             */00000370
/*                   COLUMS  4-7 IS THE CURRENT YEAR, 19__.              */00000380
/*      CARDS NA+5 - ? -- THE VT#, ED#, PAGE# CARDS,WITH A BLANK         */00000390
/*                        CARD AT THE END OF THE DECK,FOLLOW HERE.       */00000400
/*                        THESE CARDS ARE FOR THE BAD DOCUMENTS ONLY.    */00000410
/*    -CARDS-                                                            */00000420
/*      THESE ARE THE DATA CARDS FOR THE DOCUMENTS WITH ERRORS WHICH     */00000430
/*      HAVE BEEN CORRECTED.  THE DECK STARTS AND ENDS WITH A BEGIN      */00000440
/*      DOCUMENT CARD ('-' IN COLUMN 1, BLANK IN COLS. 2-80) AND SUCH    */00000450
/*      A CARD SEPARATES THE DATA FOR EACH DOCUMENT.                     */00000460
/*                                                                       */00000470
/*************************************************************************/00000480
                                                                         00000490
EDIT:   PROCEDURE OPTIONS (MAIN);                                        00000500
        DECLARE                                                          00000510
          CAT                  CHARACTER (3736) VARYING,                 00000520
          DUMMY                                                          00000530
              CHARACTER (1),                                             00000540
          (NA,N10,NT,NC,NS,N50,N60,N70,N80,N90,N100,N200)                00000550
              FIXED BINARY (31,0),                                       00000560
          NDOC                 FIXED BINARY (31,0),                      00000570
          MAST1                FILE SEQUENTIAL INPUT RECORD,             00000580
```

```
          MASTER              FILE SEQUENTIAL OUTPUT RECORD,                00000590
          V(900)              CHARACTER (10),                               00000600
          VT                                                                00000610
              CHARACTER (10),                                               00000620
        ;                                                                   00000630
                                                                            00000640
                                                                            00000650
/*  COPY THE OLD MASTER DATA SET TO A NEW MASTER DATA SET, LEAVING          */00000660
/*  OFF THE DOCUMENTS WHOSE RECORDS HAVE ERRORS.                            */00000670
                                                                            00000680
                                                                            00000690
        GET FILE (SYSIN) EDIT (NA,DUMMY) (F(4),X(75),A(1));                 00000700
LOOP1:  DO NC = 1 TO NA;                                                    00000710
        GET FILE (SYSIN) EDIT (V(NC),DUMMY) (A(10),X(69),A(1));             00000720
        END LOOP1;                                                          00000730
        ON ENDFILE (MAST1) GO TO DATA;                                      00000740
READ:   READ FILE (MAST1) INTO (CAT);                                       00000750
        GET STRING (CAT) EDIT (VT) (A(10));                                 00000760
LOOP2:  DO NC = 1 TO NA;                                                    00000770
        IF VT = V(NC) THEN GO TO READ;                                      00000780
        END LOOP2;                                                          00000790
        WRITE FILE (MASTER) FROM (CAT);                                     00000800
        PUT LIST (VT);                                                      00000810
        GO TO READ;                                                         00000820
DATA:   CLOSE FILE (MAST1);                                                 00000830
        GET FILE (SYSIN) EDIT (NA,N10,NT,NC,NS,N50,N60,N70,N80,N90,         00000840
            N100,N200,DUMMY) (12 F(3),X(43),A(1));                          00000850
        GET FILE (SYSIN) EDIT (NDOC,DUMMY) (F(5),X(7 ),  (1));              00000860
                                                                            00000870
                                                                            00000880
/*  THIS BLOCK IS ALMOST IDENTICAL TO THE MASTER PROGRAM.  IT WILL          */00000890
/*  PROCESS DATA CARDS FOR A DOCUMENT AND WRITE A RECORD ON THE             */00000900
/*  NEW MASTER DATA SET FOR THAT DOCUMENT.                                  */00000910
                                                                            00000920
                                                                            00000930
WORK:   BEGIN;                                                              00000940
        DECLARE                                                             00000950
          AUTHOR(NA)          CHARACTER (50) VARYING,                       00000960
          AVAILABILITY        CHARACTER ((N70-69)*132) VARYING,             00000970
          BD                  CHARACTER (1) INITIAL ('¬'),                  00000980
          BEG                                                               00000990
              CHARACTER (1),                                                00001000
          BT                  CHARACTER (1) INITIAL ('<'),                  00001010
          BUREAU_NO           CHARACTER ((N60-59)*132/2) VARYING,           00001020
          CARD_NO                                                           00001030
              FIXED BINARY (31,0),                                          00001040
          CODE_NO                                                           00001050
                  ACTER (17),                                               00001060
          CON     (_NO        CHARACTER ((N50-49)*132/3) VARYING,           00001070
          CORP_AUTHOR(NC-29) CHARACTER (132) VARYING,                       00001080
          CR                  CHARACTER (1) INITIAL ('|'),                  00001090
          DESCRIP             CHARACTER ((N100-99)*132) VARYING,            00001100
          DESC_NOTE           CHARACTER ((N90-89)*132-13) VARYING,          00001110
          F(NDOC)             CHARACTER (20),                               00001120
          ED_NO               CHARACTER (20),                               00001130
          ET                  CHARACTER (1) INITIAL ('>'),                  00001140
```

```
GRANT_NO            CHARACTER ((N50-49)*132/3) VARYING,          00001150
                                                                 00001160
I                                                                00001170
    FIXED BINARY (31,0),                                         00001180
IDENT               CHARACTER ((N200-199)*132) VARYING,          00001190
IND                 FIXED BINARY (31,0),                         00001200
JOUR_CIT            CHARACTER ((N80-79)*132) VARYING,            00001210
K                                                                00001220
    FIXED BINARY (31,0),                                         00001230
L                                                                00001240
    FIXED BINARY (31,0),                                         00001250
LINE_NO                                                          00001260
    FIXED BINARY (31,0),                                         00001270
LONG                CHARACTER (660) VARYING,                     00001280
L_AUTHOR(NA)        FIXED BINARY (31,0),                         00001290
L_AVAILABILITY                                                   00001300
    FIXED BINARY (31,0),                                         00001310
L_BUREAU_NO                                                      00001320
    FIXED BINARY (31,0),                                         00001330
L_CONTRACT_NO                                                    00001340
    FIXED BINARY (31,0),                                         00001350
L_CORP_AUTHOR(NC-29) FIXED BINARY (31,0),                        00001360
L_DESCRIP                                                        00001370
    FIXED BINARY (31,0),                                         00001380
L_DESC_NOTE                                                      00001390
    FIXED BINARY (31,0),                                         00001400
L_GRANT_NO                                                       00001410
    FIXED BINARY (31,0),                                         00001420
L_IDENT                                                          00001430
    FIXED BINARY (31,0),                                         00001440
L_JOUR_CIT                                                       00001450
    FIXED BINARY (31,0),                                         00001460
L_PAGE                                                           00001470
    FIXED BINARY (31,0),                                         00001480
L_PA_NO                                                          00001490
    FIXED BINARY (31,0),                                         00001500
L_PRICE                                                          00001510
    FIXED BINARY (31,0),                                         00001520
L_PUB_DATE                                                       00001530
    FIXED BINARY (31,0),                                         00001540
L_REPORT_NO                                                      00001550
    FIXED BINARY (31,0),                                         00001560
L_SPONSOR_AG(NS-39) FIXED BINARY (31,0),                         00001570
L_TITLE                                                          00001580
    FIXED BINARY (31,0),                                         00001590
N                                                                00001600
    FIXED BINARY (31,0),                                         00001610
NO_AUTHOR                                                        00001620
    FIXED BINARY (31,0),                                         00001630
NO_CORP_AUTHOR                                                   00001640
    FIXED BINARY (31,0),                                         00001650
NO_DES                                                           00001660
    FIXED BINARY (31,0),                                         00001670
NO_RECS                                                          00001680
    FIXED BINARY (31,0),                                         00001690
NO_SPONSOR                                                       00001700
    FIXED BINARY (31,0),
```

```
          PAGE                                                                  00001710
              CHARACTER (16) VARYING,                                           00001720
          PAGE_NO             CHARACTER (4),                                    00001730
          PA_NO                                                                 00001740
              CHARACTER (2),                                                    00001750
          PG(NDOC)            CHARACTER (4),                                    00001760
          PRICE               CHARACTER ((N50-49)*132/2) VARYING,               00001770
          PUB_DATE                                                              00001780
              CHARACTER (7),                                                    00001790
          REPORT_NO           CHARACTER ((N60-59)*132/2) VARYING,               00001800
          SEASON              CHARACTER (3),                                    00001810
          SEP                 CHARACTER (2) INITIAL ('||'),                     00001820
          SEPERATE                                                              00001830
              ENTRY (FIXED BINARY (31,0),FIXED BINARY (31,0),                   00001840
                     FIXED BINARY (31,0),,),                                    00001850
          SPONSOR_AG(NS-39) CHARACTER (132) VARYING,                            00001860
          STR                 CHARACTER (80),                                   00001870
          TAB                                                                   00001880
              FIXED BINARY (31,0),                                              00001890
          TB                  CHARACTER (1) INITIAL ('¬'),                      00001900
          TITLE               CHARACTER ((NT-19)*132) VARYING,                  00001910
          VT_NO                                                                 00001920
              CHARACTER (10),                                                   00001930
          YEAR                                                                  00001940
              CHARACTER (4),                                                    00001950
          ;                                                                     00001960
        OPEN FILE (ERRE) PRINT STREAM OUTPUT LINESIZE(125),                     00001970
             FILE (CARDS) INPUT RECORD;                                         00001980
        ON ENDFILE (CARDS) GO TO STOP;                                          00001990
        ON CONVERSION BEGIN;                                                    00002000
          PUT FILE (ERRE) EDIT ('CONVERSION ERROR IN DOCUMENT ',VT)             00002010
            (SKIP(2),A(29),A(10));                                              00002020
          GO TO SKIP;                                                           00002030
          END;                                                                  00002040
        GET FILE (SYSIN) EDIT (SEASON,YEAR,DUMMY)                               00002050
             (A(3),A(4),X(72),A(1));                                            00002060
ED_LP:  DO I = 1 TO NDOC;                                                       00002070
           GET FILE (SYSIN) EDIT (V(I),E(I),PG(I),DUMMY)                        00002080
                (A(10),A(20),A(4),X(45),A(1));                                  00002090
           IF V(I) = '          ' THEN GO TO ASGN;                              00002100
           K = I;                                                               00002110
        END ED_LP;                                                              00002120
ASGN:   NO_RECS = 0;                                                            00002130
        READ FILE (CARDS) IGNORE (1);                                           00002140
START:  READ FILE (CARDS) INTO (STR);                                           00002150
        GET STRING (STR) EDIT (VT_NO,LINE_NO,CARD_NO,CODE_NO)                   00002160
          (A(10),F(3),F(1),A(17));                                              00002170
        IND = 0;                                                                00002180
        IMP = 2;                                                                00002190
        TAB = 0;                                                                00002200
        VT = VT_NO;                                                             00002210
        N = INDEX(CODE_NO,CR);                                                  00002220
        IF N = 0 THEN GO TO TEST;                                               00002230
        CODE_NO = SUBSTR(CODE_NO,1,N-1);                                        00002240
TEST:   IF LINE_NO ¬= 1 THEN GO TO ERR;                                         00002250
        IF CARD_NO ¬= 1 THEN GO TO ERR;                                         00002260
```

```
          IF (SUBSTR(CODE_NO,6,2)<40) | (SUBSTR(CODE_NO,6,2)>52)          00002270
            THEN DO;                                                      00002280
                 PUT FILE (ERRF) EDIT ('CODE NO. IS OUT OF BOUNDS')       00002290
                   (SKIP(2),A(25));                                       00002300
                 GO TO SKIP;                                              00002310
                 END;                                                     00002320
          NO_AUTHOR = 0;                                                  00002330
          CALL SEPERATE(10,N10,1,TLE_SECT,LONG);                          00002340
SRCH:     N = INDEX(LONG,';');                                            00002350
          IF N ¬= 0 THEN DO;                                              00002360
                         NO_AUTHOR = NO_AUTHOR + 1;                       00002370
                         IF NO_AUTHOR > NA THEN GO TO OFF_LNTH;           00002380
                         AUTHOR(NO_AUTHOR) = SUBSTR(LONG,1,N-1);          00002390
                         L_AUTHOR(NO_AUTHOR) = N - 1;                     00002400
                         IF (N-1) > 50 THEN GO TO OFF_LNTH;               00002410
                         LONG = SUBSTR(LONG,N+2);                         00002420
                         GO TO SRCH;                                      00002430
                         END;                                             00002440
          NO_AUTHOR = NO_AUTHOR + 1;                                      00002450
          IF NO_AUTHOR > NA THEN GO TO OFF_LNTH;                          00002460
          N = INDEX(LONG,CR);                                             00002470
          IF N = 0 THEN GO TO MISSCR;                                     00002480
          AUTHOR(NO_AUTHOR) = SUBSTR(LONG,1,N-1);                         00002490
          L_AUTHOR(NO_AUTHOR) = N - 1;                                    00002500
          IF (N-1) > 50 THEN GO TO OFF_LNTH;                              00002510
TLE_SECT:                                                                 00002520
          CALL SEPERATE (20,NT,2,CORP_SECT,LONG);                         00002530
          N = INDEX(LONG,CR);                                             00002540
          IF N = 0 THEN GO TO MISSCR;                                     00002550
          TITLE = SUBSTR(LONG,1,N-1);                                     00002560
          L_TITLE = N - 1;                                                00002570
          I = 0;                                                          00002580
          IDENT = TITLE;                                                  00002590
RET:      N = INDEX(IDENT,BT);                                            00002600
          L = INDEX(IDENT,ET);                                            00002610
          IF (N=0) & (L=0) & (I¬=0) THEN GO TO CORP_SECT;                 00002620
          IF (N=0) | (L=0)                                                00002630
            THEN DO;                                                      00002640
                 PUT FILE (ERRF) EDIT                                     00002650
                   ('SHORT TITLE MARKS ARE NOT CORRECT IN DOCUMENT ',VT)  00002660
                   (SKIP(2),A(46),A(10));                                 00002670
                 GO TO SKIP;                                              00002680
                 END;                                                     00002690
          I = I + 1;                                                      00002700
          IDENT = SUBSTR(IDENT,L+2);                                      00002710
          GO TO RET;                                                      00002720
CORP_SECT:                                                                00002730
          NO_CORP_AUTHOR = 0;                                             00002740
CP_LP:    DO I = 30 TO NC;                                                00002750
          CALL SEPERATE (I,I,3,SPON_SECT,LONG);                           00002760
          NO_CORP_AUTHOR = NO_CORP_AUTHOR + 1;                            00002770
          N = INDEX(LONG,CR);                                             00002780
          IF N = 0 THEN GO TO MISSCR;                                     00002790
          CORP_AUTHOR(NO_CORP_AUTHOR) = SUBSTR(LONG,1,N-1);               00002800
          L_CORP_AUTHOR(NO_CORP_AUTHOR) = N - 1;                          00002810
          IF (N-1) > 132 THEN GO TO OFF_LNTH;                             00002820
```

```
          END CP_LP;                                                    00002830
SPON_SECT:                                                              00002840
          NO_SPONSOR = 0;                                               00002850
SP_LP:    DO I = 40 TO NS;                                              00002860
          CALL SEPERATE (I,I,4,PRICE_SECT,LONG);                        00002870
          NO_SPONSOR=NO_SPONSOR + 1;                                    00002880
          N = INDEX(LONG,CR);                                           00002890
          IF N = 0 THEN GO TO MISSCR;                                   00002900
          SPONSOR_AG(NO_SPONSOR) = SUBSTR(LONG,1,N-1);                  00002910
          L_SPONSOR_AG(NO_SPONSOR) = N - 1;                             00002920
          IF (N-1) > 132 THEN GO TO OFF_LNTH;                           00002930
          END SP_LP;                                                    00002940
PRICE_SECT:                                                             00002950
          L_CONTRACT_NO,L_GRANT_NO,L_PA_NO = 0;                         00002960
          CALL SEPERATE (50,N50,5,REP_BUR_SECT,LONG);                   00002970
          N = INDEX(LONG,TB);                                           00002980
          IF N = 0 THEN GO TO MISSCR;                                   00002990
          IF N = 1 THEN GO TO OFF_LNTH;                                 00003000
          PRICE = SUBSTR(LONG,1,N-1);                                   00003010
          L_PRICE = N - 1;                                              00003020
          IF (N-1) > ((N50-49)*132/2) THEN GO TO OFF_LNTH;              00003030
          LONG = SUBSTR(LONG,N+1);                                      00003040
          N = INDEX(LONG,TB);                                           00003050
          IF N = 0 THEN GO TO MISSCR;                                   00003060
          IF N = 1 THEN GO TO MISS1;                                    00003070
          CONTRACT_NO = SUBSTR(LONG,1,N-1);                             00003080
          L_CONTRACT_NO = N - 1;                                        00003090
          IF (N-1) > ((N50-49)*132/3) THEN GO TO OFF_LNTH;              00003100
MISS1:    LONG = SUBSTR(LONG,N+1);                                      00003110
          N = INDEX(LONG,TB);                                           00003120
          IF N = 0 THEN GO TO MISSCR;                                   00003130
          IF N = 1 THEN GO TO MISS2;                                    00003140
          GRANT_NO = SUBSTR(LONG,1,N-1);                                00003150
          L_GRANT_NO = N - 1;                                           00003160
          IF (N-1) > ((N50-49)*132/3) THEN GO TO OFF_LNTH;              00003170
MISS2:    LONG = SUBSTR(LONG,N+1);                                      00003180
          N = INDEX(LONG,CR);                                           00003190
          IF N = 0 THEN GO TO MISSCR;                                   00003200
          IF N = 1 THEN GO TO REP_BUR_SECT;                             00003210
          PA_NO = SUBSTR(LONG,1,N-1);                                   00003220
          L_PA_NO = N - 1;                                              00003230
          IF (N-1) >= 2 THEN GO TO OFF_LNTH;                            00003240
REP_BUR_SECT:                                                           00003250
          L_REPORT_NO,L_BUREAU_NO = 0;                                  00003260
          CALL SEPERATE(60,N60,6,AVL_SECT,LONG);                        00003270
          N = INDEX(LONG,TB);                                           00003280
          IF N = 0 THEN GO TO MISSCR;                                   00003290
          IF N = 1 THEN GO TO BUR;                                      00003300
          REPORT_NO = SUBSTR(LONG,1,N-1);                               00003310
          L_REPORT_NO = N - 1;                                          00003320
          IF (N-1) > ((N60-59)*132/2) THEN GO TO OFF_LNTH;              00003330
BUR:      LONG = SUBSTR(LONG,N+1);                                      00003340
          N = INDEX(LONG,CR);                                           00003350
          IF N = 0 THEN GO TO MISSCR;                                   00003360
          IF N = 1 THEN GO TO AVL_SECT;                                 00003370
          BUREAU_NO = SUBSTR(LONG,1,N-1);                               00003380
```

```
          L_BUREAU_NO = N - 1;                                          00003390
          IF (N-1) > ((N60-59)*132/2) THEN GO TO OFF_LNTH;              00003400
AVL_SECT:                                                               00003410
          L_AVAILABILITY = 0;                                           00003420
          CALL SEPERATE (70,N70,7,JC_SECT,LONG);                        00003430
          N = INDEX(LONG,CR);                                           00003440
          IF N = 0 THEN GO TO MISSCR;                                   00003450
          AVAILABILITY = SUBSTR(LONG,1,N-1);                            00003460
          L_AVAILABILITY = N - 1;                                       00003470
JC_SECT:                                                                00003480
          L_JOUR_CIT = 0;                                               00003490
          CALL SEPERATE (80,N80,8,PUB_SECT,LONG);                       00003500
          N = INDEX(LONG,CR);                                           00003510
          IF N = 0 THEN GO TO MISSCR;                                   00003520
          JOUR_CIT = SUBSTR(LONG,1,N-1);                                00003530
          L_JOUR_CIT = N - 1;                                           00003540
PUB_SECT:                                                               00003550
          L_PUB_DATE,L_PAGE,L_DESC_NOTE = 0;                            00003560
          CALL SEPERATE (90,N90,9,DESC_SECT,LONG);                      00003570
          N = INDEX(LONG,TB);                                           00003580
          IF N = 0 THEN GO TO MISSCR;                                   00003590
          IF N = 1 THEN DO;                                             00003600
                        PUB_DATE = '       ';                           00003610
                        L_PUB_DATE = 7;                                 00003620
                        GO TO GP;                                       00003630
                        END;                                            00003640
          PUB_DATE = SUBSTR(LONG,1,N-1);                                00003650
          L_PUB_DATE = N - 1;                                           00003660
          IF (N-1) ¬= 7 THEN GO TO OFF_LNTH;                            00003670
GP:       LONG = SUBSTR(LONG,N + 1);                                    00003680
          N = INDEX(LONG,TB);                                           00003690
          IF N = 0 THEN GO TO MISSCR;                                   00003700
          IF N = 1 THEN DO;                                             00003710
                        PAGE = '   ';                                   00003720
                        L_PAGE = 3;                                     00003730
                        GO TO DS;                                       00003740
                        END;                                            00003750
          PAGE = SUBSTR(LONG,1,N-1);                                    00003760
          L_PAGE = N - 1;                                               00003770
          IF (N-1) > 16 THEN GO TO OFF_LNTH;                            00003780
DS:       LONG = SUBSTR(LONG,N+1);                                      00003790
          N = INDEX(LONG,CR);                                           00003800
          IF N = 0 THEN GO TO MISSCR;                                   00003810
          IF N = 1 THEN GO TO DESC_SECT;                                00003820
          DESC_NOTE=SUBSTR(LONG,1,N-1);                                 00003830
          L_DESC_NOTE = N - 1;                                          00003840
DESC_SECT:                                                              00003850
          L_DESCRIP = 0;                                                00003860
          IND = 10;                                                     00003870
          CALL SEPERATE (100,N100,10,ID_SECT,DESCRIP);                  00003880
          NO_DES = 0;                                                   00003890
RET2:     N = INDEX(DESCRIP,CR);                                        00003900
          IF N = 0 THEN GO TO MISSCR;                                   00003910
          DESCRIP = SUBSTR(DESCRIP,1,N-1);                              00003920
          L_DESCRIP = N - 1;                                            00003930
          IF NO_DES = 1 THEN GO TO SEARCH;                              00003940
```

```
ID_SECT:                                                        00003950
        L_IDENT = 0;                                            00003960
        IND = 11;                                               00003970
        IMP = 0;                                                00003980
        CALL SEPERATE (200,N200,11,SEARCH,LONG);                00003990
        IMP = 1;                                                00004000
RETRN:  IF LONG = '' THEN GO TO SEARCH;                         00004010
        N = INDEX(LONG,CR);                                     00004020
        IF N = 0 THEN GO TO MISSCR;                             00004030
        IDENT = SUBSTR(LONG,1,N-1);                             00004040
        L_IDENT = N - 1;                                        00004050
        GO TO SEARCH;                                           00004060
ERR:    DO;                                                     00004070
        PUT FILE (ERRE) EDIT                                    00004080
          ('MISMATCHED CARD NUMBER IN DOCUMENT ',VT)            00004090
          (SKIP(2),A(35),A(10));                                00004100
        GO TO SKIP;                                             00004110
        END ERR;                                                00004120
MISSCR: DO;                                                     00004130
        PUT FILE (ERRE) EDIT                                    00004140
          ('MISSING END OF LINE OR TAB CHARACTER IN ',VT)       00004150
          (SKIP(2),A(37),A(10));                                00004160
        IF IMP = 2 THEN GO TO SKIP;                             00004170
        IF IMP=1 THEN READ FILE (CARDS) IGNORE (1);             00004180
        GO TO START;                                            00004190
        END;                                                    00004200
OFF_LNTH: DO;                                                   00004210
          PUT FILE (ERRE) EDIT                                  00004220
            ('LENGTH OF VARIABLE OR DIMENSION IS OFF IN ',VT)   00004230
            (SKIP(2),A(42),A(10));                              00004240
          GO TO SKIP;                                           00004250
          END;                                                  00004260
ERROR:  IF SUBSTR(VT_NO,1,1) ¬= BD THEN GO TO BAD;              00004270
        IF (IND = 10) & (VT_NO ¬= VT) THEN DO;                  00004280
                                            IMP = 0;            00004290
                                            L_IDENT = 0;        00004300
                                            NO_DES = 1;         00004310
                                            GO TO RET2;         00004320
                                            END;                00004330
        IF (IND = 11) & (VT_NO ¬= VT) THEN GO TO RETRN;         00004340
        BAD:   PUT FILE (ERRE) SKIP(2);                         00004350
               PUT FILE (ERRE) EDIT                             00004360
                 ('DOCUMENT NO. DOES NOT MATCH ON ONE CARD IN ',VT) 00004370
                 (A(43),A(10));                                 00004380
               GO TO SKIP;                                      00004390
NO_INF: PUT FILE (ERRE) EDIT                                    00004400
          ('EXPECTED INFORMATION MISSING FROM DOCUMENT ',VT)    00004410
          (SKIP(2),A(34),A(10));                                00004420
                                                                00004430
                                                                00004440
             /* THESE STATEMENTS SKIP TO THE BEGINNING OF THE   00004450
               NEXT DOCUMENT.                                   */ 00004460
                                                                00004470
             SKIP: READ FILE (CARDS) INTO (STR);                00004480
                   GET STRING (STR) EDIT (BEG)  (A(1));         00004490
                   IF BEG = BD THEN GO TO START;                00004500
```

```
                    GO TO SKIP;                                                 00004510
                                                                                00004520
                                                                                00004530
SEARCH: DO I = 1 TO K;                                                          00004540
          IF V(I) = VT THEN DO;                                                 00004550
                            ED_NO = E(I);                                       00004560
                            PAGE_NO = PG(I);                                    00004570
                            GO TO PGE;                                          00004580
                            END;                                                00004590
        END SEARCH;                                                             00004600
        PUT FILE (ERRE) EDIT ('NO MATCH MADE ON ED NUMBER FOR ',VT)             00004610
          (A(31),A(10));                                                        00004620
        IF IMP=1 THEN READ FILE (CARDS) IGNORE (1);                             00004630
        GO TO START;                                                            00004640
PGE:    L = NO_AUTHOR + NO_CORP_AUTHOR + NO_SPONSOR + 17;                       00004650
        PUT STRING (CAT) EDIT (VT,CODE_NO,ED_NO,SEASON,YEAR,PAGE_NO,(           00004660
          AUTHOR(I) DO I = 1 TO NO_AUTHOR),TITLE,(CORP_AUTHOR(I) DO I =00004670
          1 TO NO_CORP_AUTHOR),(SPONSOR_AG(I) DO I = 1 TO NO_SPONSOR),          00004680
          PRICE,CONTRACT_NO,GRANT_NO,PA_NO,REPORT_NO,BUREAU_NO,                 00004690
          AVAILABILITY,JOUR_CIT,PUB_DATE,PAGE,DESC_NOTE,DESCRIP,IDENT,          00004700
          SEP,NO_AUTHOR,(L_AUTHOR(I) DO I=1 TO NO_AUTHOR),L_TITLE,              00004710
          NO_CORP_AUTHOR,(L_CORP_AUTHOR(I) DO I=1 TO NO_CORP_AUTHOR),           00004720
          NO_SPONSOR,(L_SPONSOR_AG(I) DO I=1 TO NO_SPONSOR),L_PRICE,            00004730
          L_CONTRACT_NO,L_GRANT_NO,L_PA_NO,L_REPORT_NO,L_BUREAU_NO,             00004740
          L_AVAILABILITY,L_JOUR_CIT,L_PUB_DATE,L_PAGE,L_DESC_NOTE,              00004750
          L_DESCRIP,L_IDENT)                                                    00004760
          (A(10),A(17),A(20),A(3),A(4),A(4),(NO_AUTHOR) A,A(L_TITLE),           00004770
          (NO_CORP_AUTHOR) A,(NO_SPONSOR) A,A(L_PRICE),A(L_CONTRACT_NO)00004780
          ,A(L_GRANT_NO),A(L_PA_NO),A(L_REPORT_NO),A(L_BUREAU_NO),              00004790
          A(L_AVAILABILITY),A(L_JOUR_CIT),                                      00004800
          A(L_PUB_DATE),A(L_PAGE),A(L_DESC_NOTE),A(L_DESCRIP),                  00004810
          A(L_IDENT),A(2),(L) F(4));                                            00004820
        WRITE FILE (MASTER) FROM (CAT);                                         00004830
        PUT FILE (SYSPRINT) SKIP(2);                                            00004840
        PUT FILE (SYSPRINT) EDIT (CAT) (A(LENGTH(CAT)));                        00004850
        NO_RECS = NO_RECS + 1;                                                  00004860
        IF IMP=1 THEN READ FILE (CARDS) IGNORE (1);                             00004870
        GO TO START;                                                            00004880
SEPERATE: PROCEDURE (K,L,IND,LBL_NXT,BIG);                                      00004890
        DECLARE                                                                 00004900
          K                                                                     00004910
              FIXED BINARY (31,0),                                              00004920
          L                                                                     00004930
              FIXED BINARY (31,0),                                              00004940
          I                                                                     00004950
              FIXED BINARY (31,0),                                              00004960
          J                                                                     00004970
              FIXED BINARY (31,0),                                              00004980
          BIG                 CHARACTER (660) VARYING,                          00004990
          LITTLE                                                                00005000
              CHARACTER (66),                                                   00005010
          IND                                                                   00005020
              FIXED BINARY (31,0),                                              00005030
          LBL_NXT                                                               00005040
              LABEL (CORP_SECT,SPON_SECT,PRICE_SECT,REP_BUR_SECT,               00005050
                     AVL_SECT,JC_SECT,PUB_SECT,DESC_SECT,ID_SECT,               00005060
```

```
                    SEARCH,TLE_SECT),                                    00005070
        ;                                                                00005080
      BIG = '';                                                          00005090
LOOP: DO I = K TO L;                                                     00005100
      IF TAB = 1 THEN GO TO LN_CK;                                       00005110
      READ FILE (CARDS) INTO (STR);                                      00005120
      GET STRING (STR) EDIT (VT_NO,LINE_NO,CARD_NO)                      00005130
        (A(10),F(3),F(1));                                               00005140
      IF VT_NO ¬= VT THEN GO TO ERROR;                                   00005150
LN_CK: IF LINE_NO = I THEN GO TO CD_CK;                                  00005160
      IF (IND=2)|(IND=9)|(IND=10)|(IND=5)                                00005170
                              THEN DO;                                   00005180
                                   IF I = K THEN GO TO NO_INF;           00005190
                                   TAB = 1;                              00005200
                                   RETURN;                               00005210
                                   END;                                  00005220
                              ELSE DO;                                   00005230
                                   TAB = 1;                              00005240
                                   IF I = K THEN GO TO LBL_NXT;          00005250
                                   RETURN;                               00005260
                                   END;                                  00005270
CD_CK: DO J = 1 TO 2;                                                    00005280
      IF J = 1 THEN GO TO NO_RD;                                         00005290
      READ FILE (CARDS) INTO (STR);                                      00005300
      GET STRING (STR) EDIT (VT_NO,LINE_NO,CARD_NO)                      00005310
        (A(10),F(3),F(1));                                               00005320
      IF VT_NO ¬= VT THEN GO TO ERROR;                                   00005330
      IF LINE_NO ¬= I THEN DO;                                           00005340
                           TAB = 1;                                      00005350
                           RETURN;                                       00005360
                           END;                                          00005370
NO_RD: IF CARD_NO ¬= J THEN GO TO ERR;                                   00005380
      GET STRING (STR) EDIT (LITTLE) (X(14),A(66));                      00005390
      BIG = BIG || LITTLE;                                               00005400
      END CD_CK;                                                         00005410
      TAB = 0;                                                           00005420
END_LP: END LOOP;                                                        00005430
      RETURN;                                                            00005440
      END SEPERATE;                                                      00005450
STOP: PUT DATA (NO_RECS);                                                00005460
      IF NO_RECS ¬= K THEN PUT FILE (ERRE) EDIT                          00005470
        ('NOT ALL DOCUMENTS WERE PROCESSED, THERE SHOULD BE ',K)         00005480
        (SKIP(2),A(50),F(4));                                            00005490
      CLOSE FILE (MASTER), FILE (ERRE);                                  00005500
      CLOSE FILE (CARDS);                                                00005510
      END WORK;                                                          00005520
      END EDIT;                                                          00005530
```

SUBJECT QUARTERLY

```
/*****************************************************************************/00000010
/*                                                                           */00000020
/*                    Q U A R T E R L Y   S U B J E C T   I N D E X          */00000030
/*                                                                           */00000040
/*                                                                           */00000050
/*  PURPOSE:  THIS PROGRAM PRODUCES A QUARTERLY SUBJECT INDEX                */00000060
/*            PRINTOUT FOR BOTH AIM AND ARM.  THE MASTER DATA SET            */00000070
/*            THAT HAS BEEN GENERATED FOR THE QUARTER IS THE MAIN            */00000080
/*            INPUT FOR THE PROGRAM.  THE NECESSARY DATA IS TAKEN            */00000090
/*            FROM MASTER AND PUT ON A TEMPORARY DISK FILE.  WE THEN         */00000100
/*            CALL FOR THE SORT/MERGE, WHICH SORTS THE RECORDS ON            */00000110
/*            THE DISK AND PUTS THE SORTED RECORDS ON ANOTHER DISK           */00000120
/*            FILE.  THE SORTED DATA IS PRINTED UNDER A FORMAT WHICH         */00000130
/*            THE VOCATIONAL CENTER SPECIFIES AND THIS IS THE INDEX          */00000140
/*            PRINTOUT FOR THE PUBLICATION.  NOW, EACH RECORD ON THE         */00000150
/*            SORTED DISK FILE IS SHORTENED AND SAVED ON MAGNETIC            */00000160
/*            TAPE FOR THE ANNUAL INDEX.                                     */00000170
/*                                                                           */00000180
/*                                                                           */00000190
/*  DEFINITION OF VARIABLES:                                                 */00000200
/*     1. BT -- CHARACTER DENOTING THE BEGINNING OF A SHORT TITLE            */00000210
/*               SEGMENT (<).                                                */00000220
/*     2. COL -- THE NUMBER OF COLUMNS OF OUTPUT IN THE INDEX                */00000230
/*               PRINTOUT.                                                   */00000240
/*     3. DESC -- DESCRIPTOR STRING FROM THE MASTER TAPE.                    */00000250
/*     4. DUMMY -- STRING USED FOR INPUTING AND OUTPUTING TAPE AND           */00000260
/*                 DISK RECORDS.                                             */00000270
/*     5. ED_NO -- ED OR ACCESSION NUMBER.                                   */00000280
/*     6. ET -- CHARACTER DENOTING THE END OF A SHORT TITLE SEGMENT.         */00000290
/*              (>)                                                          */00000300
/*     7. HEAD -- A STRING WHICH HEADS EVERY PAGE OF THE INDEX.  IT          */00000310
/*                GIVES AIM OR ARM, THE SEASON, AND THE YEAR.  (FOR          */00000320
/*                EXAMPLE, AIM/FALL 69)                                      */00000330
/*     8. IDENT -- IDENTIFIER STRING FROM THE MASTER TAPE.                   */00000340
/*     9. IHESRTA -- THE PL/I CALL TO SORT/MERGE.                            */00000350
/*    10. LENTH -- AN ARRAY USED FOR STORING LENGTHS OF STRINGS.             */00000360
/*    11. LH -- LENGTH OF -HEAD- (NO. 7).                                    */00000370
/*    12. LINEH1 -- THE LINE NUMBER IN THE INDEX PRINTOUT ON WHICH           */00000380
/*                  -HEAD- AND THE PAGE NUMBER APPEAR ON THE TITLE           */00000390
/*                  PAGE.                                                    */00000400
/*    13. LINEH2 -- SAME AS LINEH1 EXCEPT IT'S FOR ALL SUCCEEDING            */00000410
/*                  PAGES OF THE INDEX                                       */00000420
/*    14. LINEP1 -- THE LINE NUMBER IN THE INDEX PRINTOUT ON WHICH           */00000430
/*                  THE WORD 'PAGE' APPEARS ABOVE EVERY COLUMN.  THIS        */00000440
/*                  IS FOR THE TITLE PAGE ONLY.                              */00000450
/*    15. LINEP2 -- SAME AS LINEP1 EXCEPT IT'S FOR ALL SUCCEEDING            */00000460
/*                  PAGES OF THE INDEX.                                      */00000470
/*    16. LINET -- THE LINE NUMBER IN THE INDEX PRINOUT ON WHICH THE         */00000480
/*                 TITLE APPEARS.  WE DO NOT PUT A TITLE ON THE TITLE        */00000490
/*                 PAGE.                                                     */00000500
/*    17. LTH -- LENGTH OF THE TITLE OF THE INDEX.                           */00000510
/*    18. L_DESC -- LENGTH OF THE DESCRIPTOR STRING.                         */00000520
/*    19. L_IDENT -- LENGTH OF THE IDENTIFIER STRING.                        */00000530
/*    20. L_TITLE -- LENGTH OF THE TITLE OF THE DOCUMENT.                    */00000540
/*    21. MARG -- LEFT-HAND MARGIN OF THE PAGE.                              */00000550
/*    22. MASTER -- NAME OF THE MASTER TAPE.                                 */00000560
/*    23. MG -- ARRAY OF MARGINS FOR THE COLUMNS IN THE INDEX                */00000570
/*              PRINTOUT.                                                    */00000580
/*    24. OUT -- ARRAY OF STARRED IDENTIFIERS AND DESCRIPTORS FOR A          */00000590
/*               DOCUMENT.                                                   */00000600
/*    25. PAGE -- PAGE NUMBER OF THE INDEX PRINTOUT.                         */00000610
/*    26. PAGE_NO -- RESUME PAGE FROM MASTER.                                */00000620
/*    27. PUB_DATE -- PUBLICATION DATE FROM MASTER.                          */
```

```
/*    28. OUE -- TELLS US WHETHER THE INDEX TITLE PAGE IS ON THE      */00000630
/*                LEFT OR RIGHT SIDE OF THE PUBLICATION.              */00000640
/*    29. RETCDE -- CODE RETURNED BY THE SORT PROGRAM.  '0' MEANS     */00000650
/*                SORT WENT WELL; '16' MEANS SORT HAD TROUBLE.        */00000660
/*    30. RFIELD -- DESCRIBES THE RECORDS BEING SORTED FOR THE SORT   */00000670
/*                PROGRAM.                                            */00000680
/*    31. SFIELD -- DESCRIBES THE PART(S) OF THE RECORD TO BE SORTED  */00000690
/*                FOR THE SORT PROGRAM.                               */00000700
/*    32. SIZE -- THE NUMBER OF LINES THAT THE INDEX PRINTOUT MAY     */00000710
/*                HAVE PER PAGE.  IT COUNTS FROM LINE 1 AND STOPS AT  */00000720
/*                THE LAST ALLOWABLE LINE ON WHICH INDEX PRINTOUT IS  */00000730
/*                ALLOWED.                                            */00000740
/*    33. SORTIN -- DEFINES THE DISK SUBSET FILE WITH THE UNSORTED    */00000750
/*                RECORDS.                                            */00000760
/*    34. SORTOUT -- DEFINES THE DISK SUBSET FILE WITH THE SORTED     */00000770
/*                RECORDS.                                            */00000780
/*    35. SPACE -- THE NUMBER OF SPACES BETWEEN THE VT AND ED NUMBER  */00000790
/*                IN THE INDEX PRINTOUT.                              */00000800
/*    36. SUM1 -- THE NUMBER OF BYTES ON A MASTER RECORD BETWEEN THE  */00000810
/*                RESUME PAGE AND THE TITLE.  THIS IS REALLY THE      */00000820
/*                SUM OF THE NAMES OF THE PERSONAL AUTHORS IF THERE   */00000830
/*                ARE ANY.                                            */00000840
/*    37. SUM2 -- THE NUMBER OF BYTES ON A MASTER RECORD BETWEEN THE  */00000850
/*                TITLE AND THE PUBLICATION DATE.                     */00000860
/*    38. SUM3 -- THE NUMBER OF BYTES ON A MASTER RECORD BETWEEN THE  */00000870
/*                PUBLICATION DATE AND THE DESCRIPTORS.  THIS IS      */00000880
/*                REALLY THE SUM OF THE LENGTH  OF THE DESCRIPTIVE    */00000890
/*                NOTE AND THE LENGTH OF THE PAGE VARIABLE.           */00000900
/*    39. SZ -- THE RIGHT-HAND MARGIN OF THE INDEX PRINTOUT PAGES.    */00000910
/*    40. TITL -- THE TITLE OF THE INDEX, 'SUBJECT AND IDENTIFIER     */00000920
/*                INDEX'.                                             */00000930
/*    41. TITLE -- TITLE OF A DOCUMENT FROM THE MASTER RECORD.        */00000940
/*    42. TLE -- ARRAY OF SHORT TITLE SEGMENTS.                       */00000950
/*    43. VT_NO -- DOCUMENT NUMBER FROM MASTER TAPE.                  */00000960
/*    44. WIDTH -- WIDTH OF THE COLUMNS IN THE INDEX PRINTOUT.        */00000970
/*    45. BLK -- NUMBER OF PIECES OF WIDTH -WIDTH- THAT A KEYWORD     */00000980
/*                IS BROKEN UP INTO TO FIT IN A COLUMN IN THE INDEX   */00000990
/*                PRINTOUT.                                           */00001000
/*    46. BLKT -- NUMBER OF PIECES OF WIDTH -WIDTH-2- THAT A TITLE    */00001010
/*                OF A DOCUMENT IS BROKEN UP INTO TO FIT IN A COLUMN  */00001020
/*                IN THE INDEX PRINTOUT.                              */00001030
/*    47. COL# -- IS A TWO DIMENSIONAL ARRAY THAT REPRESENTS AN       */00001040
/*                INDEX PRINTOUT PAGE.  ITS FIRST DIMENSION GIVES THE */00001050
/*                COLUMN NUMBER OF SOME COLUMN ON A PAGE, AND ITS     */00001060
/*                SECOND DIMENSION IS A LINE NUMBER IN THE COLUMN     */00001070
/*                REPRESENTED BY THE FIRST DIMENSION.  I PUT A PAGE   */00001080
/*                OF THE INDEX INTO THIS ARRAY AND THEN PRINT THE     */00001090
/*                ARRAY.  THIS IS THE ONLY WAY TO PRINT A WHOLE PAGE  */00001100
/*                AT ONCE WHEN DATA IS ASSIGNED TO A PAGE IN VERTICAL */00001110
/*                COLUMNS AND THE COMPUTER PRINTER PRINTS             */00001120
/*                HORIZONTALLY ONLY.                                  */00001130
/*    48. IMAX -- IS THE NUMBER OF COLUMNS PLUS ONE.                  */00001140
/*    49. IMP -- INDICATES WHETHER A NEW KEYWORD IS BEING STARTED     */00001150
/*                IN THE INDEX PRINTOUT.                              */00001160
/*    50. IND -- INDICATES WHAT  COLUMN OF A PAGE IS BEING FILLED.    */00001170
/*    51. KNT -- IS A COUNT OF THE NUMBER OF LINES IN A BLOCK OF DATA */00001180
/*                BY BLOCK I MEAN, VT NUMBER, ED NUMBER  AND RESUME   */00001190
/*                PAGE ON ONE LINE AND TITLE BROKEN UP INTO -BLKT-    */00001200
/*                LINES.  ALSO, IF WE ARE LISTING THE FIRST DOCUMENT  */00001210
/*                FOR A KEYWORD, WE COUNT THE KEYWORD BROKEN UP INTO  */00001220
/*                -BLK- LINES, PLUS ONE FOR THE BLANK LINE BETWEEN    */00001230
/*                KEYWORD LISTINGS.                                   */00001240
```

```
/*     52. LN -- IS THE LAST LINE IN A PARTICULAR COLUMN THAT HAS      */00001250
/*               BEEN FILLED WITH INDEX PRINTOUT.                      */00001260
/*     53. NN -- IS A SUBSCRIPT USED FOR THE SECOND DIMENSION OF       */00001270
/*               -COL#-                                                */00001280
/*     54. OUTS -- NAME OF OUTPUT FILE WHERE THE INDEX IS PRINTED.     */00001290
/*     55. PC -- THE PIECES OF WIDTH -WIDTH- INTO WHICH A KEYWORD IS   */00001300
/*               BROKEN UP.                                            */00001310
/*     56. PREV -- KEEPS THE NAME OF THE CURRENT KEYWORD BEING LISTED  */00001320
/*                 IN THE INDEX.                                       */00001330
/*     57. PUB -- THE YEAR SECTION OF THE PUBLICATION DATE.            */00001340
/*     58. SKPH -- GIVES THE MARGIN FOR VARIABLE -HEAD- SO I KNOW      */00001350
/*                 WHERE TO POSITION IT.                               */00001360
/*     59. SKPP -- GIVES THE MARGIN FOR THE PAGE NUMBER ON THE INDEX   */00001370
/*                 PRINTOUT.                                           */00001380
/*     60. SUB -- NAME OF THE MAGNETIC TAPE DATA SET ON WHICH THE      */00001390
/*                SHORTENED RECORDS FOR THE ANNUAL INDEX ARE SAVED.    */00001400
/*     61. TLP -- THE PIECES OF WIDTH -WIDTH-2- INTO WHICH A TITLE IS  */00001410
/*                BROKEN UP.                                           */00001420
/*                                                                     */00001430
/*   INPUT TO THE PROGRAM:                                             */00001440
/*     -MASTER-, THE MASTER TAPE DATA SET                              */00001450
/*       THE MASTER TAPE DATA SET GENERATED FOR THE QUARTER IS ONE      */00001460
/*       INPUT TO THE PROGRAM.  IT IS REFERENCED BY DD CARD IN THE     */00001470
/*       JCL.                                                          */00001480
/*     -SYSIN-, INPUT ON CARDS                                         */00001490
/*       WE NEED TO INPUT TO THE PROGRAM THE OUTPUT SPECIFICATIONS     */00001500
/*       FOR THE INDEX PRINTOUT.  THE FOLLOWING LIST TELLS WHAT DATA   */00001510
/*       IS READ IN.  THE DATA IS ONE CONTINUOUS STREAM AND TAKES AS   */00001520
/*       MANY CARDS AS NECESSARY.                                      */00001530
/*       FIELD 1 IS THE PAGE NUMBER FOR THE FIRST PAGE OF THE INDEX    */00001540
/*             PRINTOUT.  F(4,0) FORMAT.                               */00001550
/*       FIELD 2 IS THE SIZE OF THE PRINTOUT PAGE.  F(4,0) FORMAT.     */00001560
/*       FIELD 3 IS THE COLUMN WIDTH.  F(4,0) FORMAT.                  */00001570
/*       FIELD 4 IS THE LEFT-HAND MARGIN OF THE PAGE. F(4,0) FORMAT.   */00001580
/*       FIELD 5 IS THE RIGHT-HAND MARGIN OF THE PAGE. F(4,0)          */00001590
/*       FIELD 6 IS THE LINE ON THE TITLE PAGE WHERE THE PAGE NUMBER   */00001600
             AND HEADING BELONG. F(4,0)                                 */00001610
          FIELD 7 IS THE LINE ON THE TITLE PAGE WHERE THE WORD 'PAGE'   */00001620
             IS WRITTEN ABOVE EACH COLUMN. F(4,0)                       */00001630
          FIELD 8 IS THE LINE ON ALL SUCCEEDING PAGES WHERE THE PAGE    */00001640
/*           NUMBER AND HEADING BELONG. F(4,0)                         */00001650
/*       FIELD 9 IS THE LINE ON ALL SUCCEEDING PAGES WHERE THE TITLE   */00001660
/*             SHOULD APPEAR.  F(4,0)                                  */00001670
/*       FIELD 10 IS THE LINE ON ALL SUCCEEDING PAGES WHERE THE WORD   */00001680
/*             'PAGE' IS WRITTEN ABOVE EACH COLUMN.  F(4,0)            */00001690
/*       FIELD 11 IS SPACE BETWEEN THE VT AND ED NUMBER IN THE         */00001700
/*             PRINTOUT.  F(4,0)                                       */00001710
/*       FIELD 12 IS THE NUMBER OF COLUMNS IN THE INDEX PRINTOUT.      */00001720
/*             F(4,0)                                                  */00001730
/*       FIELD 13 IS THE MARGIN FOR EACH COLUMN MEASURED FROM COLUMN   */00001740
/*             ONE.  FORMAT IS THE NUMBER OF COLUMNS TIMES F(4,0).     */00001750
/*       FIELD 14 IS THE LENGTH OF THE INDEX TITLE.  F(4,0)            */00001760
/*       FIELD 15 IS THE INDEX TITLE. FORMAT IS A(FIELD 14'S NO.).     */00001770
/*       FIELD 16 IS THE LENGTH OF THE HEADING.  F(4,0)                */00001780
/*       FIELD 17 IS THE HEADING (EX: FALL/ARM 69).  FORMAT IS         */00001790
/*             A(FIELD 16'S NO.).                                      */00001800
/*       FIELD 18 IS AN 'L' OR 'R'. 'L' SAYS THAT THE FIRST PAGE OF    */00001810
/*             THE INDEX WILL BE ON THE LEFT SIDE OF THE PUBLICATION.  */00001820
/*             'R' INDICATES THE RIGHT SIDE.  FORMAT A(1).             */00001830
/*                                                                     */00001840
/*   SPECIAL NOTE:                                                     */00001850
/*     THE OUTPUT FILE SHOULD BE THE PRINTER UNTIL THE VOCATIONAL      */00001860
```

```
/*    CENTER SEES AND APPROVES THE INDEX PRINTOUT.  WE DEFINE IT AS  */00001870
/*    '//OUTS DD SYSOUT=A,DCB=(RECFM=F,LRECL=133,BLKSIZE=133)'. THEN */00001880
/*    OUTS SHOULD BE DEFINED AS A TAPE.  THE OUTPUT TAPE IS THEN     */00001890
/*    LISTED ON THE 360/50 FOR AN UPPER AND LOWER CASE PRINTOUT. THE */00001900
/*    JCL FOR THIS 50 RUN IS IN THE MISCELLANEOUS PROGRAM SECTION.   */00001910
/*                                                                   */00001920
/*******************************************************************/ 00001930
                                                                       00001940
                                                                       00001950
SUBIND: PROCEDURE OPTIONS (MAIN);                                      00001960
        DECLARE                                                        00001970
            BT                  CHARACTER (1) INITIAL ('<'),           00001980
            COL                                                        00001990
                FIXED BINARY (31,0),                                   00002000
            DESC                                                       00002010
                CHARACTER (660) VARYING,                               00002020
            DISECT                                                     00002030
                ENTRY,                                                 00002040
            DUMMY               CHARACTER (3736) VARYING,              00002050
            ED_NO               CHARACTER (20),                        00002060
            ET                  CHARACTER (1) INITIAL ('>'),           00002070
            HEAD                CHARACTER (13) VARYING,                00002080
            I                                                          00002090
                FIXED BINARY (31,0),                                   00002100
            IDENT               CHARACTER (396) VARYING,               00002110
            IHESRTA                                                    00002120
                ENTRY(CHARACTER(48),CHARACTER(34),FIXED BINARY(31,0),  00002130
                    FIXED BINARY(15,0)),                               00002140
            J                                                          00002150
                FIXED BINARY (31,0),                                   00002160
            L                                                          00002170
                FIXED BINARY (31,0),                                   00002180
            LENTH(10)           FIXED BINARY (31,0),                   00002190
            LH                  FIXED BINARY (31,0),                   00002200
            LINEH1              FIXED BINARY (31,0),                   00002210
            LINEH2              FIXED BINARY (31,0),                   00002220
            LINEP1              FIXED BINARY (31,0),                   00002230
            LINEP2              FIXED BINARY (31,0),                   00002240
            LINET               FIXED BINARY (31,0),                   00002250
            LTH                                                        00002260
                FIXED BINARY (31,0),                                   00002270
            L_DESC              FIXED BINARY (31,0),                   00002280
            L_IDENT             FIXED BINARY (31,0),                   00002290
            L_TITLE             FIXED BINARY (31,0),                   00002300
            M                                                          00002310
                FIXED BINARY (31,0),                                   00002320
            MARG                FIXED BINARY (31,0),                   00002330
            MASTER                                                     00002340
                FILE SEQUENTIAL UNBUFFERED INPUT RECORD,               00002350
            MG(3)               FIXED BINARY (31,0),                   00002360
            N                                                          00002370
                FIXED BINARY (31,0),                                   00002380
            NUMB                FIXED BINARY (31,0),                   00002390
            OUT(15) CHAR(200) VAR,                                     00002400
            PAGE                                                       00002410
                FIXED BINARY (31,0),                                   00002420
            PAGE_NO             CHARACTER (4),                         00002430
            PUB_DATE                                                   00002440
                CHARACTER (7),                                         00002450
            QUE                 CHARACTER (1),                         00002460
            RETCDE                                                     00002470
                FIXED BINARY (15,0),                                   00002480
```

```
          RFIELD                                                        00002490
              CHARACTER (34),                                           00002500
          SFIELD                                                        00002510
              CHARACTER (48),                                           00002520
          SIZE                                                          00002530
              FIXED BINARY (31,0),                                      00002540
          SORTIN                                                        00002550
              FILE SEQUENTIAL RECORD,                                   00002560
          SORTOUT                                                       00002570
              FILE SEQUENTIAL RECORD,                                   00002580
          SPACE              FIXED BINARY (31,0),                       00002590
          SUM1                                                          00002600
              FIXED BINARY (31,0),                                      00002610
          SUM2                                                          00002620
              FIXED BINARY (31,0),                                      00002630
          SUM3                                                          00002640
              FIXED BINARY (31,0),                                      00002650
          SZ                 FIXED BINARY (31,0),                       00002660
          TITL                                                          00002670
              CHARACTER (50) VARYING,                                   00002680
          TITLE              CHARACTER (396) VARYING,                   00002690
          TLE(3)                                                        00002700
              CHARACTER (396) VARYING,                                  00002710
          VT_NO                                                         00002720
              CHARACTER (10) VARYING,                                   00002730
          WIDTH                                                         00002740
              FIXED BINARY (31,0),                                      00002750
          ;                                                             00002760
                                                                        00002770
          OPEN FILE (MASTER),                                           00002780
               FILE (ERRF) OUTPUT PRINT STREAM LINESIZE (125),          00002790
               FILE (SORTIN) OUTPUT;                                    00002800
          ON ENDFILE (MASTER) GO TO SORT;                               00002810
                                                                        00002820
                                                                        00002830
     /*  READ A RECORD FROM THE MASTER TAPE.  SEARCH FOR THE DIVIDER  (||) 00002840
         IN THE RECORD.                                                 */00002850
                                                                        00002860
                                                                        00002870
     READ_MASTER:                                                       00002880
          READ FILE (MASTER) INTO (DUMMY);                              00002890
          N = INDEX(DUMMY,'||');                                        00002900
          IF N = 0                                                      00002910
            THEN DO:                                                    00002920
                 VT_NO = SUBSTR(DUMMY,1,10);                            00002930
                 PUT FILE (ERRF) EDIT                                   00002940
                   ('MISSING DIVIDER IN DOCUMENT ',VT_NO)               00002950
                   (SKIP(2),A(28),A(10));                               00002960
                 GO TO READ_MASTER;                                     00002970
                 END;                                                   00002980
                                                                        00002990
                                                                        00003000
     /*  FIND THE NUMBER OF POSITIONS FROM THE RESUME PAGE TO THE TITLE, 00003010
         SO THAT THE TITLE CAN BE PULLED OFF OF MASTER.  -SUM1- IS THIS 00003020
         VALUE.  ALSO, GET THE TITLE LENGTH.                            */00003030
                                                                        00003040
                                                                        00003050
          SUM1 = 0;                                                     00003060
          GET STRING (DUMMY) EDIT (NUMB,(LENTH(I) DO I = 1 TO NUMB),    00003070
            L_TITLE)  (X(N+1),F(4),(NUMB+1) F(4));                      00003080
          N = N + 9 + (NUMB * 4);                                       00003090
          IF NUMB¬= 0 THEN DO;                                          00003100
```

```
                    LOOP1: DO I = 1 TO NUMB;                                00003110
                           SUM1 = SUM1 + LENTH(I);                          00003120
                           END LOOP1;                                       00003130
                    END;                                                    00003140
-------------------------------------------------------------------------------00003150
                                                                            00003160
    /*  FIND THE NUMBER OF POSITIONS FROM THE TITLE TO THE PUBLICATION      00003170
        DATE SO THAT THE DATE CAN BE PULLED OFF OF MASTER.  -SUM2- IS       00003180
        THIS VALUE.                                                        */00003190
                                                                            00003200
-------------------------------------------------------------------------------00003210
        SUM2 = 0;                                                           00003220
LOOP2:  DO J = 1 TO 2;                                                      00003230
        GET STRING (DUMMY) EDIT (NUMB,(LENTH(I) DO I=1 TO NUMB))            00003240
          (X(N),F(4),(NUMB) F(4));                                          00003250
        N = N + 4 + (4 * NUMB);                                             00003260
        IF NUMB ¬= 0 THEN DO;                                               00003270
                    INNER: DO L = 1 TO NUMB;                                00003280
                           SUM2 = SUM2 + LENTH(L);                          00003290
                           END INNER;                                       00003300
                    END;                                                    00003310
        END LOOP2;                                                          00003320
LOOP3:  DO I = 1 TO 8;                                                      00003330
        GET STRING (DUMMY) EDIT (NUMB)                                      00003340
          (X(N),F(4));                                                      00003350
        SUM2 = SUM2 + NUMB;                                                 00003360
        N = N + 4;                                                          00003370
        END LOOP3;                                                          00003380
        N = N + 4;                                                          00003390
                                                                            00003400
                                                                            00003410
    /*  FIND THE NUMBER OF POSITIONS FROM THE DATE TO THE IDENTIFIERS       00003420
        AND DESCRIPTORS SO THAT THESE STRINGS CAN BE PULLED OFF OF MASTER.00003430
        -SUM3- IS THIS VALUE.  ALSO, GET THESE STRINGS' LENGTHS.          */00003440
-------------------------------------------------------------------------------00003450
                                                                            00003460
        SUM3 = 0;                                                           00003470
LOOP3A: DO I = 1 TO 2;                                                      00003480
        GET STRING (DUMMY) EDIT (NUMB) (X(N),F(4));                         00003490
        SUM3 = SUM3 + NUMB;                                                 00003500
        N = N + 4;                                                          00003510
        END LOOP3A;                                                         00003520
        GET STRING (DUMMY) EDIT (L_DESC,L_IDENT) (X(N),2 F(4));             00003530
                                                                            00003540
                                                                            00003550
    /*  WITH ALL THIS FROM MASTER, THE NECESSARY DATA CAN BE RETRIEVED      00003560
        FROM THE TAPE.                                                    */00003570
                                                                            00003580
                                                                            00003590
        GET STRING (DUMMY) EDIT (VT_NO,ED_NO,PAGE_NO,TITLE,PUB_DATE,       00003600
                                 DESC,IDENT)                                00003610
            (A(10),X(17),A(20),X(7),A(4),X(SUM1),A(L_TITLE),X(SUM2),        00003620
            A(7),X(SUM3),A(L_DESC),A(L_IDENT));                             00003630
        PUT SKIP (4);                                                       00003640
        PUT EDIT (VT_NO,ED_NO,PAGE_NO,TITLE,PUB_DATE,DESC,IDENT) (A);       00003650
                                                                            00003660
                                                                            00003670
    /*  FORM THE SHORT TITLE FROM THE TITLE PULLED OFF OF MASTER.         */00003680
-------------------------------------------------------------------------------00003690
                                                                            00003700
        I = 0;                                                              00003710
RETRN:  N = INDEX(TITLE,BT);                                                00003720
```

```
          M = INDEX(TITLE,ET);                                              00003730
          IF (N=0) & (M=0) THEN GO TO LOOP4;                                00003740
          I = I + 1;                                                        00003750
          TLE(I) = SUBSTR(TITLE,N+1,M-N-1);                                 00003760
          IF SUBSTR(TITLE,M+1,1) ¬= ' ' THEN TLE(I) = TLE(I) ||             00003770
                                              SUBSTR(TITLE,M+1,1);          00003780
          TITLE = SUBSTR(TITLE,M+2);                                        00003790
          GO TO RETRN;                                                      00003800
LOOP4:    DO J = 1 TO I;                                                    00003810
          IF J = 1 THEN TITLE = TLE(1);                                     00003820
                    ELSE TITLE = TITLE || ' ' || TLE(J);                    00003830
          END LOOP4;                                                        00003840
          PUT DATA (TITLE);                                                 00003850
                                                                            00003860
                                                                            00003870
   /*  BREAK UP THE DESCRIPTOR AND IDENTIFIER STRINGS INTO THE INDIVIDUAL   00003880
       KEYWORDS.  STORE THE STARRED ONES IN -OUT-.                          */00003890
                                                                            00003900
                                                                            00003910
WHOLE:    I = 0;                                                            00003920
          CALL DISECT(DESC,I,OUT);                                          00003930
          IF L_IDENT = 0 THEN GO TO LOOP5;                                  00003940
          DESC = IDENT;                                                     00003950
          CALL DISECT(DESC,I,OUT);                                          00003960
                                                                            00003970
                                                                            00003980
   /*  PUT THE STARRED KEYWORDS WITH THEIR CORRESPONDING INFORMATION        00003990
       ON THE DISK SUBSET FILE IN INDIVIDUAL RECORDS PER KEYWORD.           */00004000
                                                                            00004010
                                                                            00004020
LOOP5:    DO J = 1 TO I;                                                    00004030
NONE:     PUT STRING (DUMMY) EDIT (OUT(J),VT_NO,ED_NO,PAGE_NO,PUB_DATE,    00004040
                                   TITLE)                                   00004050
            (A(200),A(10),A(20),A(4),A(7),A);                               00004060
          WRITE FILE (SORTIN) FROM (DUMMY);                                 00004070
          IDENT = OUT(J);                                                   00004080
          PUT DATA (IDENT);                                                 00004090
FL5:      END LOOP5;                                                        00004100
                                                                            00004110
          GO TO READ_MASTER;                                                00004120
                                                                            00004130
                                                                            00004140
   /*  THIS PROCEDURE BREAKS UP THE DESCRIPTOR  AND IDENTIFIER STRINGS      00004150
       INTO THE INDIVIDUAL KEYWORDS AND STORES THE STARRED ONES IN -OUT-.   00004160
       IT ALSO CHECKS THAT EACH KEYWORD FITS WITHIN THE 132 BYTE LENGTH     00004170
       MAXIMUM THAT I HAVE CHOSEN.                                          */00004180
                                                                            00004190
                                                                            00004200
DISECT: PROCEDURE (STRING,I,OUT);                                           00004210
          DECLARE                                                           00004220
            STRING              CHARACTER (660) VARYING,                    00004230
            I                   FIXED BINARY (31,0),                        00004240
            OUT(15) CHAR(200) VAR,                                          00004250
            REP                 CHARACTER (1) INITIAL ('"'),                00004260
          ;                                                                 00004270
SRC:      N = INDEX(STRING,';');                                            00004280
          IF N = 0 THEN DO;                                                 00004290
                        N = LENGTH(STRING);                                 00004300
                        N = N + 1;                                          00004310
                        END;                                                00004320
          IF N>200                                                          00004330
          THEN DO;                                                          00004340
```

```
                PUT FILE (ERRE) EDIT                                    00004350
                  ('KEYWORD TOO LARGE IN DOCUMENT ',VT_NO)              00004360
                  (SKIP(2),A(30),A(10));                                00004370
                GO TO TRN;                                              00004380
                END;                                                    00004390
            IF SUBSTR(STRING,1,1)='*' THEN DO;                          00004400
                                        I = I + 1;                      00004410
                                        OUT(I)=SUBSTR(STRING,2,N-2)||   00004420
                                               REP;                     00004430
                                        END;                            00004440
        TRN:N = INDEX(STRING,';');                                      00004450
            IF N = 0 THEN RETURN;                                       00004460
            STRING = SUBSTR(STRING,N+2);                                00004470
            GO TO SRC:                                                  00004480
            END DISECT;                                                 00004490
                                                                        00004500
                                                                        00004510
     /*  MAKE THE PROPER PREPARATIONS AND CALL SORT.  PRINT THE SORT RETURN00004520
         CODE; '0' INDICATES A GOOD SORT.                               */00004530
                                                                        00004540
                                                                        00004550
    SORT:   CLOSE FILE (SORTIN),                                        00004560
                FILE (MASTER);                                          00004570
            SFIELD=' SORT FIELDS=(5,200,CH,A,208,7,CH,A),SIZE=E9999 ';  00004580
            RFIELD=' RECORD TYPE=V,LENGTH=(641,,,245) ';                00004590
            CALL IHESRTA(SFIELD,RFIELD,45056,RETCDE);                   00004600
            PUT FILE (SYSPRINT) EDIT ('RETURN CODE = ',RETCDE)          00004610
              (SKIP(3),A(14),F(3));                                     00004620
                                                                        00004630
                                                                        00004640
     /*  READ IN THE PAGE SPECIFICATIONS FOR THE QUARTERLY INDEX PRINTOUT. 00004650
                                                                       */00004660
                                                                        00004670
                                                                        00004680
            GET FILE (SYSIN) EDIT (PAGE,SIZE,WIDTH,MARG,SZ,LINEH1,LINEP1, 00004690
              LINEH2,LINET,LINEP2,SPACE,COL,(MG(N) DO N = 1 TO COL),LTH, 00004700
              TITL,LH,HEAD,QUE)                                         00004710
              (13 F(4),(COL) F(4),A(LTH),F(4),A(LH),A(1));              00004720
                                                                        00004730
                                                                        00004740
        /*  THE PRINTING OF THE INDEX AND THE SHORTENING OF THE OUTPUT FILE 00004750
            ARE ACCOMPLISHED IN THIS BLOCK                              */00004760
                                                                        00004770
                                                                        00004780
        WORK:   BEGIN;                                                  00004790
                DECLARE                                                 00004800
                  BLK                                                   00004810
                    FIXED BINARY (31,0),                                00004820
                  BLKT                                                  00004830
                    FIXED BINARY (31,0),                                00004840
                  BREAK             ENTRY(,FIXED BINARY (31,0),,),      00004850
                  COL#(COL,SIZE-LINEP2-1)  CHARACTER (WIDTH+3),         00004860
                  IMAX                                                  00004870
                    FIXED BINARY (31,0),                                00004880
                  IMP               FIXED BINARY (31,0),              00004890
                  IND                                                   00004900
                    FIXED BINARY (31,0),                                00004910
                  K                                                     00004920
                    FIXED BINARY (31,0),                                00004930
                  KK                FIXED BINARY (31,0),              00004940
                  KNT               FIXED BINARY (31,0),              00004950
                  LL                FIXED BINARY (31,0),              00004960
```

```
          LN
               FIXED BINARY (31,0),                                    00004970
          NN                   FIXED BINARY (31,0),                    00004980
          OUTS                                                         00004990
               FILE,                                                   00005000
          PC(14)               CHARACTER (WIDTH) VARYING,              00005010
          PREV CHAR(200) VAR INITIAL('99'),                            00005020
          PUB                                                          00005030
               CHARACTER (2),                                          00005040
          SKPH                 FIXED BINARY (31,0),                    00005050
          SKPP                 FIXED BINARY (31,0),                    00005060
          SUB                                                          00005070
               FILE SEQUENTIAL RECORD,                                 00005080
          TLP(14)              CHARACTER (WIDTH) VARYING,              00005090
          ;                                                            00005100
                                                                       00005110
                                                                       00005120
                                                                       00005130
     /*  INITIALIZE VARIABLES AS NEEDED FOR PRINTING.  OPEN THE OUTPUT  00005140
          FILE WHICH RECEIVES THE INDEX PRINTOUT.                      */00005150
                                                                       00005160
                                                                       00005170
          ON ENDFILE (SORTOUT) GO TO SHORTEN;                          00005180
          OPEN FILE (SORTOUT),                                         00005190
               FILE (OUTS) OUTPUT STREAM PRINT PAGESIZE(SIZE+4)        00005200
          ;                                                            00005210
                                                                       00005220
     BLKOT: DO I = 1 TO SIZE-LINEP2-1;                                 00005230
          DO J = 1 TO COL;                                             00005240
          COL#(J,I) = ' ';                                             00005250
          END BLKOT;                                                   00005260
          K = PAGE;                                                    00005270
          IND =1;                                                      00005280
          IMAX = COL + 1;                                              00005290
          LN = LINEP1;                                                 00005300
          NN = 0;                                                      00005310
          IF OUE = 'L' THEN DO;                                        00005320
                            SKPP = MARG;                               00005330
                            SKPH = SZ - LH + 1;                        00005340
                            END;                                       00005350
                       ELSE DO;                                        00005360
                            SKPH = MARG;                               00005370
                            SKPP = SZ - 3;                             00005380
                            END;                                       00005390
                                                                       00005400
     /*  PRINT THE HEADING FOR THE TITLE PAGE.                         */00005410
                                                                       00005420
                                                                       00005430
          PUT FILE (OUTS) EDIT (PAGE,HEAD)                             00005440
            (PAGE,LINE(LINEH1),COLUMN(SKPP),F(4),SKIP(0),COLUMN(SKPH), 00005450
            A(LH));                                                    00005460
          PUT FILE (OUTS) SKIP (LINEP1-LINEH1);                        00005470
          LP8: DO I = 1 TO COL;                                        00005480
               PUT FILE (OUTS) EDIT ('PAGE')                           00005490
                 (SKIP(0),COLUMN(MG(I)+WIDTH-1),A(4));                 00005500
               END LP8;                                                00005510
          PUT FILE (OUTS) SKIP(1);                                     00005520
                                                                       00005530
                                                                       00005540
     /*  READ IN A RECORD FROM THE SORTED DISK SUBSET FILE.  BREAK UP THE 00005550
          RECORD INTO THE INDIVIDUAL INFORMATION.  SEARCH FOR THE MARK IN 00005560
          THE KEYWORD THAT INDICATES WHERE THE BLANKS ON THE RIGHT BEGIN AND00005570
          CUT THE KEYWORD AT THIS POINT.  BREAK UP THE TITLE INTO STRINGS 00005580
```

```
      THAT FIT WITHIN THE COLUMN WIDTH SPECIFIED                          */00005590
                                                                           00005600
                                                                           00005610
   READ:  READ FILE (SORTOUT) INTO (DUMMY);                                00005620
          L = LENGTH(DUMMY);                                               00005630
          GET STRING (DUMMY) EDIT (IDENT,VT_NO,ED_NO,PAGE_NO,PUB_DATE,     00005640
               TITLE)(A(200),A(10),A(20),A(4),A(7),A(L-241));              00005650
          N = INDEX(IDENT,'|');                                            00005660
          IDENT = SUBSTR(IDENT,1,N-1);                                     00005670
          CALL BREAK (TITLE,WIDTH-2,TLP,BLKT);                             00005680
                                                                           00005690
                                                                           00005700
   /* IF THE RECORD IS THE START OF DATA FOR A NEW KEYWORD, THEN BREAK      00005710
      UP THE KEYWORD INTO STRINGS OF LENGTH THAT MATCH THE COLUMN WIDTH.    00005720
      INDICATE A NEW KEYWORD'S START WITH -IMP-.  INITIALIZE -KNT-,THE     00005730
      NUMBER OF LINES IN THIS BLOCK OF DATA.                              */00005740
                                                                           00005750
                                                                           00005760
          IF IDENT ¬= PREV                                                 00005770
            THEN DO;                                                       00005780
                 PREV = IDENT;                                             00005790
                 IF (N-1) > WIDTH THEN CALL BREAK(IDENT,WIDTH,PC,BLK);     00005800
                                  ELSE DO;                                 00005810
                                       BLK = 1;                            00005820
                                       PC(1) = IDENT;                      00005830
                                       END;                                00005840
                 KNT = BLK + 2 + BLKT;                                     00005850
                 IMP = 1;                                                  00005860
                 END;                                                      00005870
                                                                           000 5880
                                                                           000   0
   /* IF THE SAME KEYWORD IS CONTINUING,SET VALUES IN -IMP- AND -KNT-      00005900
      FOR THIS SITUATION.                                                */00005910
                                                                           00005920
                                                                           00005930
            ELSE DO;                                                       00005940
                 KNT = BLKT + 1;                                           00005950
                 IMP = 0;                                                  00005960
                 END;                                                      00005970
                                                                           00005980
                                                                           00005990
   /* IF THE BLOCK OF DATA CANNOT FIT INTO THE CURRENT COLUMN BEING        00006000
      FILLED WITHOUT EXCEDING THE PAGE LENGTH, A NEW COLUMN MUST BE        00006010
      STARTED.                                                           */00006020
                                                                           00006030
                                                                           00006040
          IF (LN + KNT) > SIZE THEN GO TO HEADR;                           00006050
                                                                           00006060
                                                                           00006070
   /* IF THIS IS A NEW KEYWORD, PRINT ITS NAME IN THE INDEX.              */00006080
                                                                           00006090
                                                                           00006100
   RETURN: IF IMP = 1 THEN DO;                                             00006110
                           NN = NN + 1;                                    00006120
                           LP6: DO I = 1 TO BLK;                           00006130
                                COL#(IND,NN) = PC(I);                      00006140
                                NN = NN + 1;                               00006150
                                END LP6;                                   00006160
                           END;                                            00006170
                                                                           00006180
                                                                           00006190
   /* PRINT THE DOCUMENT NUMBER AND OTHER DATA FOR THE DOCUMENT BEING     00006200
```

```
        LISTED UNDER THE KEYWORD.                                          */00006210
                                                                            00006220
                                                                            00006230
            PUT STRING (COL#(IND,NN)) EDIT (VT_NO,ED_NO,PAGE_NO)            00006240
              (A(10),X(SPACE),A(20),X(WIDTH-30-SPACE-1),A(4));              00006250
            NN = NN + 1;                                                    00006260
    LP7:    DO I = 1 TO BLKT;                                               00006270
            COL#(IND,NN) = '  ' || TLP(I);                                 00006280
            NN = NN + 1;                                                    00006290
            END LP7;                                                        00006300
            LN = LN + KNT;                                                  00006310
            GO TO READ;                                                     00006320
                                                                            00006330
                                                                            00006340
     /*  A BRANCH HERE STARTS A NEW COLUMN BECAUSE THE CURRENT COLUMN IS AS00006350
         FULL AS CAN BE.  CHANGE -IND- TO THE NEXT COLUMN NUMBER ON THE     00006360
         PAGE.                                                             */00006370
                                                                            00006380
                                                                            00006390
    HEADR:  IND = IND + 1;                                                  00006400
                                                                            00006410
                                                                            00006420
         /*  IF THE LAST COLUMN ON THE PAGE IS FULL, START A NEW PAGE AS WELL 00006430
         AS A NEW COLUMN.                                                   00006440
         SET THE COLUMN NUMBER TO ONE.  PRINT OUT THE PAGE JUST COMPLETED   00006450
         ON THE OUTPUT FILE.  INCREMENT THE PAGE NUMBER OF THE INDEX AND    00006460
         PRINT A TITLE ON THE NEW PAGE.                                    */00006470
                                                                            00006480
                                                                            00006490
            IF IND = IMAX                                                   00006500
              THEN DO;                                                      00006510
                   IND = 1;                                                 00006520
                   HD1: DO LL = 1 TO SIZE-LINEP2-1;                         00006530
                        PUT FILE (OUTS) SKIP(1);                            00006540
                        LP9: DO KK = 1 TO COL;                              00006550
                             PUT FILE (OUTS) EDIT (COL#(KK,LL))             00006560
                              (SKIP(0),COLUMN(MG(KK)),A(WIDTH+3));          00006570
                             COL#(KK,LL) = ' ';                             00006580
                             END LP9;                                       00006590
                        END HD1;                                            00006600
                   PAGE = PAGE + 1;                                         00006610
                   IF SKPP =MARG THEN DO;                                   00006620
                                      SKPP = SZ - 3;                        00006630
                                      SKPH = MARG;                          00006640
                                      END;                                  00006650
                                 ELSE DO;                                   00006660
                                      SKPP = MARG;                          00006670
                                      SKPH = SZ - LH + 1;                   00006680
                                      END;                                  00006690
                  PUT FILE (OUTS) EDIT (PAGE,HEAD,TITL)                     00006700
                    (PAGE,LINE(LINEH2),COLUMN(SKPP),F(4),SKIP(0),COLUMN(    00006710
                    SKPH),A(LH),LINE(LINET),X((SZ-LTH-MARG+1)/2+MARG-1),    00006720
                   A(LTH));                                                 00006730
                  PUT FILE (OUTS) SKIP(LINEP2-LINET);                       00006740
                  LP10: DO LL = 1 TO COL;                                   00006750
                        PUT FILE (OUTS) EDIT ('PAGE')                       00006760
                          (SKIP(0),COLUMN(MG(LL)+WIDTH-1),A(4));            00006770
                        END LP10;                                           00006780
                  PUT FILE (OUTS) SKIP(1);                                  00006790
                  END;                                                      00006800
                                                                            00006810
                                                                            00006820
```

```
     /*  NEW PAGE OR NOT, INITIALIZE THE LINE COUNTERS FOR A NEW COLUMN. */00006830
                                                                          00006840
                                                                          00006850
               IF IMP = 1 THEN DO;                                        00006860
                               NN = 0;                                    00006870
                               LN = LINEP2;                               00006880
                               END;                                       00006890
                          ELSE DO;                                        00006900
                               NN = 1;                                    00006910
                               LN = LINEP2 + 1;                           00006920
                               END;                                       00006930
               IF K = PAGE THEN LN = LN + LINEP1 - LINEP2;                00006940
               GO TO RETURN;                                              00006950
                                                                          00006960
                                                                          00006970
     /*  WHEN THE SORTED FILE HAS BEEN READ COMPLETELY, PRINT OUT THE PAGE 00006980
         BEING FILLED ON THE OUTPUT FILE.  INDICATE THE LAST PAGE NUMBER OF00006990
         INDEX, AND CLOSE THE OUTPUT FILE.                               */00007000
                                                                          00007010
                                                                          00007020
     SHORTEN:                                                             00007030
               DO I = 1 TO SIZE-LINEP2-1;                                 00007040
               PUT FILE (OUTS) SKIP(1);                                   00007050
               LP11: DO J = 1 TO COL;                                     00007060
                     PUT FILE (OUTS) EDIT (COL#(J,I))                     00007070
                       (SKIP(0),COLUMN(MG(J)),A(WIDTH+3));                00007080
                     END LP11;                                            00007090
               END SHORTEN;                                               00007100
               PUT FILE (OUTS) PAGE;                                      00007110
               PUT FILE (SYSPRINT) EDIT ('LAST PAGE USED IS ',PAGE)       00007120
                 (SKIP(3),A(18),F(4));                                    00007130
               CLOSE FILE (OUTS),                                         00007140
                     FILE (SORTOUT);                                      00007150
                                                                          00007160
                                                                          00007170
     /*  SHORTEN THE RECORDS ON SORTED SUBSET FILE AND FORM A NEW SUBSET   00007180
         FILE ON TAPE.                                                   */00007190
                                                                          00007200
                                                                          00007210
               ON ENDFILE (SORTOUT) GO TO STOP;                           00007220
               OPEN FILE (SORTOUT),                                       00007230
                    FILE (SUB) OUTPUT;                                    00007240
               PUT FILE (SYSPRINT) PAGE;                                  00007250
     RD:       READ FILE (SORTOUT) INTO (DUMMY);                          00007260
               GET STRING (DUMMY) EDIT (IDENT,VT_NO,PUB)                  00007270
                     (A(200),A(10),X(29),A(2));                           00007280
               VT_NO = SUBSTR(VT_NO,4,3) || SUBSTR(VT_NO,8,3);            00007290
               PUT STRING (DUMMY) EDIT (IDENT,PUB,VT_NO)                  00007300
                     (A(200),A(2),A(6));                                  00007310
               WRITE FILE (SUB) FROM (DUMMY);                             00007320
               PUT EDIT(DUMMY)(SKIP(1),A(208));                           00007330
               GO TO RD;                                                  00007340
     STOP:     CLOSE FILE (SUB);                                          00007350
                                                                          00007360
                                                                          00007370
     /*  THIS PROCEDURE TAKES A STRING (-BREAKE-) AND BREAKS IT UP INTO    00007380
         PIECES OF LENGTH -LNTH-.  THE PIECES ARE STORED IN -BROKE- AND    00007390
         -NU- INDICATES HOW MANY PIECES IT TOOK.                         */00007400
                                                                          00007410
                                                                          00007420
     BREAK:  PROCEDURE(BREAKE,LNTH,BROKE,NU);                             00007430
               DECLARE                                                    00007440
```

```
          BREAKE              CHARACTER (396) VARYING,                  00007450
          LNTH                FIXED BINARY (31,0),                      00007460
          TEMP                CHARACTER (396) VARYING,                  00007470
          BROKE(*)            CHARACTER (WIDTH) VARYING,                00007480
          NU                  FIXED BINARY (31,0),                      00007490
          L                   FIXED BINARY (31,0),                      00007500
          SUM1                FIXED BINARY (31,0),                      00007510
        ;                                                               00007520
        NU = 0;                                                         00007530
INIT:   SUM1 = 0;                                                       00007540
        TEMP = BREAKE;                                                  00007550
SRCH:   L = INDEX(TEMP,' ');                                            00007560
        IF L = 0                                                        00007570
          THEN DO;                                                      00007580
               L = LENGTH(TEMP);                                        00007590
               SUM1 = SUM1 + L;                                         00007600
               IF SUM1 > LNTH                                           00007610
                 THEN DO;                                               00007620
                      NU = NU + 1;                                      00007630
                      BROKE(NU) = SUBSTR(BREAKE,1,SUM1-L-1);            00007640
                      NU = NU + 1;                                      00007650
                      BROKE(NU) = SUBSTR(BREAKE,SUM1-L+1);              00007660
                      RETURN;                                           00007670
                      END;                                              00007680
                 ELSE DO;                                               00007690
                      NU = NU + 1;                                      00007700
                      BROKE(NU) = SUBSTR(BREAKE,1,SUM1);                00007710
                      RETURN;                                           00007720
                      END;                                              00007730
               END;                                                     00007740
        IF L > LNTH                                                     00007750
          THEN DO;                                                      00007760
               PUT FILE (ERRE) EDIT                                     00007770
                 ('TROUBLE IN BREAK WITH ',VT_NO)                       00007780
                 (SKIP(2),A(22),A(10));                                 00007790
               GO TO READ;                                              00007800
               END;                                                     00007810
SUM:    SUM1 = SUM1 + (L - 1);                                          00007820
        IF SUM1 <=LNTH THEN DO;                                         00007830
                            TEMP = SUBSTR(TEMP,L+1);                    00007840
                            SUM1 = SUM1 + 1;                            00007850
                            GO TO SRCH;                                 00007860
                            END;                                        00007870
        NU = NU + 1;                                                    00007880
        BROKE(NU) = SUBSTR(BREAKE,1,SUM1-L);                            00007890
        BREAKE = SUBSTR(BREAKE,SUM1+2-L);                               00007900
        SUM1 = 0;                                                       00007910
        GO TO SUM;                                                      00007920
        END BREAK;                                                      00007930
                                                                        00007940
        END WORK;                                                       00007950
                                                                        00007960
        CLOSE FILE (ERRE), FILE (SORTOUT);                              00007970
        END SUBIND;                                                     00007980
```

AUTHOR QUARTERLY

```
FIRST: PROCEDURE OPTIONS(MAIN);                                          00000010
/*************************************************************************/00000020
/*                                                                       */00000030
/*          Q U A R T E R L Y    A U T H O R    I N D E X                 */00000040
/*-----------------------------------------------------------------------*/00000050
/*                                                                       */00000070
/* PURPOSE: EXTRACT DATA FROM MASTER TAPE TO BE USED FOR QUARTERLY       */00000080
/*           INDEX, ANNUAL INDEX, AND DATA BASE TO SIMPLIFY FINDING      */00000090
/*           PUBLICATION INFORMATION REGARDING A GIVEN AUTHOR.           */00000100
/*           NOTE: AUTHOR'S NAMES ARE CAPITALIZED IN THE OUTPUT DATA     */00000110
/*           VARIABLES REQUIRING DEFINITIONS ARE DEFINED WITHIN THE      */00000120
/*           PROGRAM.                                                    */00000130
/*                                                                       */00000140
/*************************************************************************/00000150
DCL BUFFER CHAR(3736) VAR,                                               00000160
    IHESRTA                                                              00000170
       ENTRY(CHAR(48),CHAR(34),FIXED BINARY(31,0),FIXED BINARY(15,0)),   00000180
       RETCDE FIXED BINARY(15,0),                                        00000190
       SFIELD CHAR(48),                                                  00000200
       RFIELD CHAR(34),                                                  00000210
     BEGIN_SHORT CHAR(1) INITIAL('<'),                                   00000220
     END_SHORT CHAR(1) INITIAL('>'),                                     00000230
       SHORT_TITLE CHAR(396) VARYING,                                    00000240
    VT# CHAR(10),                                                        00000250
    ED# CHAR(20),                                                        00000260
    NAME CHAR(132),                                                      00000270
     UPC CHAR(132) INITIAL((132)' '),                                    00000280
    RESUME_PG CHAR(4),                                                   00000290
       PER_AUT CHARACTER(132),                                           00000300
     TITLE CHARACTER(396) VARYING,                                       00000310
     CORP_AUT CHARACTER(528) VARYING,                                    00000320
    (I,#_PER_AUT,LEN_P_AUT(2),TITLE_LEN,#_COR_AUT,LEN_C_AUT(5),#SA,      00000330
        LSA(2),J,LP1,LC,LG,LPA,LP2,LB,LEN_AVAIL,LEN_J_C,LEN_PUB,         00000340
      LEN_PG#,J1,J2) FIXED BINARY(31,0),                                 00000350
       PUB_DATE CHARACTER(100) VARYING,                                  00000360
     PAGE CHARACTER(10) VARYING,                                         00000370
    BUF2 CHAR(560) VAR;                                                  00000380
                                                                         00000390
  /*                                                                     00000400
        CREATE  AUTHOR  FILE                                             00000410
  */                                                                     00000420
  CREATE: BEGIN;                                                         00000430
   OPEN FILE(SYSPRINT) LINESIZE(120);                                    00000440
  OPEN FILE(PERSON) TITLE('SORTIN') RECORD OUTPUT;                       00000450
   ENTER: READ FILE(MASTER) INTO(BUFFER);                                00000460
    ON ENDFILE(MASTER) GOTO FINISH;                                      00000470
        LEN_C_AUT=0 ;                                                    00000480
        LEN_P_AUT=0 ;                                                    00000490
       LSA=0;                                                            00000500
  /*                                                                     00000510
        FIND BEGINNING OF LENGTHS                                        00000520
  */                                                                     00000530
         I=INDEX(BUFFER,'||');                                           00000540
       IF I>0 THEN GO TO AUTHOR;                                         00000550
              ELSE DO;                                                   00000560
  /* SIGNAL FOR LENGTHS IS MISSING - TERMINATE */                        00000570
                   PUT LIST('INDEX IS WRONG',I);                         00000580
                   GO TO FINISH; END;                                    00000590
 AUTHOR: ;                                                               00000600
  /* FIND PER. AUTHORS LENGTH */                                         00000610
     GET STRING(BUFFER) EDIT(#_PER_AUT)(X(I+2),F(4));                    00000620
       I=I+6;                                                            00000630
         IF #_PER_AUT>2 THEN DO;
```

```
                                        PUT EDIT(BUFFER)(SKIP,A);                00000640
                PUT LIST('  # PERSONAL AUTHORS WRONG',#_PER_AUT);                00000650
            #_PER_AUT=0;   END;                                                  00000660
            IF #_PER_AUT=2 THEN GET STRING(BUFFER) EDIT(LEN_P_AUT(1),            00000670
              LEN_P_AUT(2))(X(I),2 F(4)); ELSE                                   00000680
            IF #_PER_AUT=1 THEN GET STRING(BUFFER) EDIT(LEN_P_AUT(1))            00000690
              (X(I),F(4));                                                       00000700
      I=I+#_PER_AUT*4;                                                           00000710
   /* FIND TITLE LENGTH AND CORP. AUTHORS */                                     00000720
      GET STRING(BUFFER) EDIT(TITLE_LEN,#_COR_AUT)(X(I),2 F(4));                 00000730
         I=I+8;                                                                  00000740
          IF #_COR_AUT>5 THEN DO;                                                00000750
                                        PUT EDIT(BUFFER)(SKIP,A);                00000760
                PUT LIST('  # CORP. AUTHOR WRONG',#_COR_AUT);                    00000770
                #_COR_AUT=0;      END;                                           00000780
       IF #_COR_AUT>0 THEN                                                       00000790
         GET STRING(BUFFER) EDIT((LEN_C_AUT(J)  DO J = 1 TO #_COR_AUT))          00000800
              (X(I), 5 F(4));                                                    00000810
  I=I+#_COR_AUT*4;                                                           00000820
   /* FIND SPACE TO AVAILABILITY INFO. */                                        00000830
        GET STRING(BUFFER) EDIT(#SA)(X(I),F(4));                                 00000840
         I=I+4;                                                                  000008 0
      IF #SA>3 THEN DO;                                                          000008 0
                                        PUT EDIT(BUFFER)(SKIP,A);                000008 0
                PUT LIST('  SPON. AG. WRONG',#SA);                              000008 0
                #SA=0;   END;                                                    000008 0
      IF #SA>0 THEN                                                              00000900
        GET STRING(BUFFER) EDIT((LSA(J)  DO J= 1 TO #SA))                        00000910
          (X(I),3 F(4));                                                         00000920
   I=I+#SA*4;                                                                  00000930
        GET STRING(BUFFER) EDIT(LP1,LC,LG,LPA,LP2,LB,LEN_AVAIL,LEN_J_C           00000940
         ,LEN_PUB,LEN_PG#)(X(I),10 F(4));                                        00000950
   /* FIND DATA FOR PERSONAL AUTHOR RECORD AND PLACE IN FILE */                  00000960
    /*                                                                           00000970
      I IS THE LENGTH OF INFORMATION TO BE SKIPPED                               00000980
    */                                                                           00000990
        I=LSA(1)+LSA(2)+LP1+LC+LG+LPA+LP2+LB+LEN_AVAIL+LEN_J_C;                  00001000
      GET STRING(BUFFER) EDIT(VT#,ED#,RESUME_PG,PER_AUT,TITLE,CORP_AUT,          00001010
        PUB_DATE)(A(10),X(17),A(20),X(7),A(4),A(SUM(LEN_P_AUT)),                 00001020
        A(TITLE_LEN),A(SUM(LEN_C_AUT)),X(I),A(LEN_PUB));                         00001030
        PUB_DATE=SUBSTR(PUB_DATE,6,2);                                           00001040
   /* PULL OFF SHORT TITLE */                                                    00001050
        J2=0;                                                                    00001060
   PULL:  J=INDEX(TITLE,BEGIN_SHORT);                                            00001070
        IF J=0 THEN GO TO PUTOUT;                                                00001080
         IF SUBSTR(TITLE,J+1,1)=BEGIN_SHORT THEN J=J+1;                          00001090
        J1=INDEX(TITLE,END_SHORT);                                               00001100
         IF J1=0 THEN GO TO PUTOUT;                                              00001110
        SUBSTR(TITLE,J1,1)=SUBSTR(TITLE,J1+1,1);                                 00001120
         J1=J1+1;                                                                00001130
    IF J2=0 THEN SHORT_TITLE=SUBSTR(TITLE,J+1,J1-J-1); ELSE                      00001140
         SHORT_TITLE=SHORT_TITLE||' '||SUBSTR(TITLE,J+1,J1-J-1);                 00001150
        TITLE=SUBSTR(TITLE,J1+1);                                                00001160
         J2=J2+1;                                                                00001170
        GO TO PULL;                                                              00001180
   PUTOUT: IF J2>0 THEN DO;                                                      00001190
             TITLE=SHORT_TITLE;                                                  00001200
             TITLE_LEN=LENGTH(SHORT_TITLE); END;                                 00001210
      IF #_PER_AUT¬=0 THEN DO;                                                   00001220
    /*                                                                           00001230
        SET UP RECORD FOR AUTHOR FILE                                            00001240
    */                                                                           00001250
```

```
          NAME=SUBSTR(PER_AUT,1,LEN_P_AUT(1));                          00001260
          UNSPEC(NAME)=UNSPEC(NAME)|UNSPEC(UPC);                        00001270
   /* * *   MAKE UP FIRST PERSONAL AUTHOR RECORD   * * */               00001280
       PUT STRING(BUF2) EDIT(NAME,VT#,ED#,RESUME_PG,PUB_DATE,TITLE_LEN,  00001290
          TITLE)(A(132),A(10),A(20),A(4),A(2),F(3),A(TITLE_LEN));       00001300
        WRITE FILE(PERSON) FROM(BUF2); END;                             00001310
     IF #_PER_AUT=2 THEN DO;                                            00001320
   /*  PLACE NEXT PERSONAL AUTHOR IN RECORD */                         00001330
          NAME=SUBSTR(PER_AUT,LEN_P_AUT(1)+1,LEN_P_AUT(2));             00001340
          UNSPEC(NAME)=UNSPEC(NAME)|UNSPEC(UPC);                        00001350
          SUBSTR(BUF2,1,132)=NAME;                                      00001360
        WRITE FILE(PERSON) FROM(BUF2); END;                             00001370
     IF #_COR_AUT¬=0 THEN DO;                                           00001380
          NAME=SUBSTR(CORP_AUT,1,LEN_C_AUT(1));                         00001390
          UNSPEC(NAME)=UNSPEC(NAME)|UNSPEC(UPC);                        00001400
   /* * *   MAKE UP FIRST INSTITUTIONAL AUTHOR RECORD   * * */          00001410
       PUT STRING(BUF2) EDIT(NAME,VT#,ED#,RESUME_PG,PUB_DATE,TITLE_LEN, 00001420
          TITLE)(A(132),A(10),A(20),A(4),A(2),F(3),A(TITLE_LEN));       00001430
        WRITE FILE(PERSON) FROM(BUF2); END;                             00001440
        LC=1;                                                           00001450
        IF #_COR_AUT>1 THEN DO J=2 TO #_COR_AUT;                        00001460
          LC=LC+LEN_CAUT(J-1);                                          00001470
          NAME=SUBSTR(CORP_AUT,LC,LEN_C_AUT(J));                       00001480
          UNSPEC(NAME)=UNSPEC(NAME)|UNSPEC(UPC);                        00001490
   /* PLACE NEXT INSTITUTIONAL AUTHOR RECORD */                         00001500
          SUBSTR(BUF2,1,132)=NAME;                                      00001510
        WRITE FILE(PERSON) FROM(BUF2); END;                             00001520
         GO TO ENTER;                                                   00001530
   /*                                                                   00001540
          CLOSE FILES IN PREPARATION FOR SORT                           00001550
   */                                                                   00001560
 FINISH: CLOSE FILE(MASTER), FILE(PERSON);                              00001570
END CREATE;                                                             00001580
SFIELD=' SORT FIELDS=(5,132,CH,A,137,10,CH,A),SIZE=E300 ';              00001590
RFIELD=' RECORD TYPE=V,LENGTH=(564,,,171) ';                            00001600
CALL IHESRTA(SFIELD,RFIELD,45056,RETCDE);                               00001610
   /*    */                                                            00001620
   /*  END OF CREATION STEP  */                                         00001630
   /*    */                                                            00001640
 IF RETCDE=0 THEN BEGIN;                                                00001650
   /*     IF SORT IS NOT SUCCESSFUL SKIP THIS BLOCK, OTHERWISE PRINT INDE00001660
    */                                                                  00001670
 DCL (LNA,LN_COL,LEN_HEADING,K,PRINT_LINE,QUARTER,PG_L_OR_R,NEXT_INITIAL00001680
     ,VT_ED_SKIP                                                        00001690
    ,MARGIN,#_COLS,COL_COUNT,INDEX_PAGE,II,LEN_PAGE,INITIAL_LINE)       00001700
          FIXED BINARY(31,0),                                           00001710
      TOP CHAR(4),                                                      00001720
      HEADING CHAR(20) VAR,                                             00001730
    COL_SPACE CHAR(20) VAR,                                             00001740
      DATE CHAR(2),                                                     00001750
      PREVIOUS CHAR(132) INITIAL((132)' ');                             00001760
   /*                                                                   00001770
      OBTAIN PARAMETERS FOR THIS PRINT                                  00001780
          #_COLS = NUMBER OF PRINT COLUMNS                              00001790
          MARGIN = TOTAL PRINT SPACE IN COLUMN FOR AUTHORS NAME         00001800
          LEN_PAGE = NUMBER OF PRINT LINES                              00001810
          HEADING = WHICH QUARTERLY INDEX AND YEAR FOR EXAMPLE - AIM/FAL00001820
          TOP = INDICATOR FOR POSITIONING OF HEADING AND INDEX PAGE NUMB00001830
              IF TOP='LEFT'THEN INDEX PAGE NUMBER APPEARS ON THE LEFT   00001840
                AND HEADING APPEARS ON RIGHT                            00001850
              IF TOP='RITERREVERSE WILL OCCUR                           00001860
          COL_SPACE = CHARACTER STRING OF BLANKS TO BE PLACED BETWEEN CO00001870
```

```
          VT_ED_SKIP = NUMBER OF COLUMNS TO BE SKIPPED BETWEEN VT# AND E00001880
               IN VT,ED,RESUME PG. LINE.                                  00001890
          INDEX_PAGE = INITIAL LINE PRINTED IN THIS INDEX                 00001900
          INITIAL_LINE = LINE FOR 'PAGE' LINE ON TITLE PAGE               00001910
          NEXT_INITIAL = LINE FOR 'PAGE' LINE ON PAGES AFTER TITLE PAGE   00001920
   */                                                                     00001930
   /* INITIALIZE PARAMETERS */                                           00001940
   GET FILE(SYSIN) DATA(#_COLS,MARGIN,LEN_PAGE,HEADING,TOP,COL_SPACE,     00001950
        VT_ED_SKIP,INDEX_PAGE,INITIAL_LINE,NEXT_INITIAL) COPY;           00001960
   I=LENGTH(COL_SPACE);                                                  00001970
   LEN_HEADING=#_COLS*(MARGIN+[illegible])+8-I;                          00001980
   PRINT_LINE=LEN_HEADING+1;                                             00001990
   BEGIN;                                                                00002000
   DCL (AUTHOR_ARRAY(6),CHAR_COL(50),SURNAME,SKIP_LINE,ONE_LINE)         00002010
         CHAR(MARGIN+3),                                                 00002020
       NEW_BLOCK FIXED BINARY(31,0) INITIAL(0),                          00002030
       PAGE_TOTAL FIXED BINARY(31,0) INITIAL(0);                         00002040
   DCL PRINTER_PAGE(LEN_PAGE) CHAR(LEN_HEADING) VAR;                     00002050
       DCL AUTHOROUT CHAR(140);                                          00002060
   PAGE_TOTAL=0;                                                         00002070
   SKIP_LINE=' ';                                                        00002080
   LEN_HEADING=LENGTH(HEADING);                                          00002090
   PRINTER_PAGE='                ';                                      00002100
   COL_COUNT=1;                                                          00002110
   QUARTER=PRINT_LINE;                                                   00002120
   J1=QUARTER-LEN_HEADING;                                               00002130
   J2=QUARTER-3;                                                         00002140
   IF TOP='LEFT' THEN DO;                                                00002150
                      QUARTER=J1;                                        00002160
                      PG_L_OR_R=9;                                       00002170
                      TOP='RITE';                                        00002180
                      END;                                               00002190
                ELSE DO;                                                 00002200
                      QUARTER=9;                                         00002210
                      PG_L_OR_R=J2;                                      00002220
                      TOP='LEFT';                                        00002230
                      END;                                               00002240
   OPEN FILE(PERSON) RECORD INPUT TITLE('SORTOUT');                      00002250
   OPEN FILE(OUT) PRINT PAGESIZE(LEN_PAGE);                              00002260
   PUT FILE(OUT) PAGE EDIT(INDEX_PAGE,HEADING)(SKIP,COLUMN(PG_L_OR_R),F(3)00002270
        ,SKIP(0),COLUMN(QUARTER),A(LEN_HEADING));                        00002280
   PUT FILE(OUT) LINE(INITIAL_LINE) EDIT(('PAGE' DO II=1 TO #_COLS))     00002290
        (X(MARGIN+7),A(4),2 (X(MARGIN+I-1),A(4)));                       00002300
   ON ENDPAGE(OUT) BEGIN;                                                00002310
         DO I=1 TO LEN_PAGE-INITIAL_LINE;                                00002320
              PUT FILE(OUT) EDIT(PRINTER_PAGE(I))(SKIP,A);               00002330
         END;                                                            00002340
              INDEX_PAGE=INDEX_PAGE+1;                                   00002350
         PRINTER_PAGE='                ';                               00002360
         PAGE_TOTAL=0;                                                   00002370
         IF TOP='LEFT' THEN DO;                                          00002380
                           TOP='RITE';                                   00002390
                           PG_L_OR_R=9;                                  00002400
                           QUARTER=J1;                                   00002410
                        END;                                             00002420
                     ELSE DO;                                            00002430
                           PG_L_OR_R=J2;                                 00002440
                           QUARTER=9;                                    00002450
                           TOP='LEFT';                                   00002460
                        END;                                             00002470
         IF LEN_PAGE<=66 THEN PUT FILE(OUT) PAGE;                        00002480
                     PUT FILE(OUT) PAGE LINE(2);                         00002490
```

```
                        PUT FILE(OUT) EDIT(INDEX_PAGE,HEADING,                  00002500
                          'PERSONAL AND INSTITUTIONAL AUTHOR INDEX',            00002510
                          ('PAGE' DO II=1 TO #_COLS))(COLUMN(PG_L_OR_R),        00002520
                         F(3),SKIP(0),COLUMN(QUARTER),A(LEN_HEADING),           00002530
                          SKIP(2),                                              00002540
                          X(PRINT_LINE/2-20),A(39),SKIP(3),X(MARGIN+7),         00002550
                          A(4),2 (X(MARGIN+LENGTH(COL_SPACE)-1),A(4)));         00002560
                        COL_COUNT=1;                                            00002570
                         INITIAL_LINE=NEXT_INITIAL;                             00002580
                            END;                                                00002590
ON_CONVERSION_BEGIN:                                                            00002600
    PUT FILE(SYSPRINT) LIST('** CONVERSION E  OR ** REC= ',BUF2,' **');         00002610
    GO TO IN;                                                                   00002620
    END;                                                                        00002630
                ON ENDFILE(PERSON) BEGIN;                                       00002640
                        SIGNAL ENDPAGE(OUT);                                    00002650
                        GO TO FINI;                                             00002660
                            END;                                                00002670
    IN: READ FILE(PERSON)INTO(BUF2);                                            00002680
    LNA=0;                                                                      00002690
      GET STRING(BUF2) EDIT(NAME,VT#,ED#,RESUME_PG,DATE,TITLE_LEN,TITLE)        00002700
          (A(132),A(10),A(20),A(4),A(2),F(3),A(TITLE_LEN));                     00002710
        /*SHORTEN AND STORE AUTHOR INFORMATION FOR ANNUAL */                    00002720
        AUTHOROUT=NAME||DATE||SUBSTR(VT#,4,3)||SUBSTR(VT#,8,3);                 00002730
      PUT FILE(ANPRT) EDIT(AUTHOROUT)(SKIP,X(2),A);                             00002740
          WRITE FILE(ANNUAL) FROM(AUTHOROUT);                                   00002750
      IF NAME=PREVIOUS THEN GO TO VT_ED;                                        00002760
      PREVIOUS=NAME;                                                            00002770
    LNA=1;                                                                      00002780
      AUTHOR_ARRAY(LNA)=SKIP_LINE;                                              00002790
    /*                                                                          00002800
          BREAK NAME UP FOR PRINT                                               00002810
    */                                                                          00002820
      DO J=MARGIN, 2*MARGIN-I,J+MARGIN-I,J+MARGIN-I,J+MARGIN-I;                 00002830
        LNA=LNA+1;                                                              00002840
        SUBNAME=SUBSTR(NAME,J-MARGIN+1,MARGIN);  I=0;                           00002850
                IF SUBSTR(SUBNAME,MARGIN,1)¬=' ' THEN DO;                       00002860
                DO I=1 TO MARGIN-1;                                             00002870
                  IF SUBSTR(SUBNAME,MARGIN-I,1)=' ' THEN GO TO BREAK;           00002880
                  END;                                                          00002890
          PUT LIST(' AUTHOR NAME IS WRONG **',NAME,' **  VT# = ',VT#);          00002900
                                        END;                                    00002910
    BREAK: SUBNAME=SUBSTR(SUBNAME,1,MARGIN-I);                                  00002920
        K=INDEX(SUBNAME,'  ');                                                  00002930
        IF K=1 THEN GO TO VT_ED;                                                00002940
        IF K=0 THEN K=MARGIN-I;                                                 00002950
        PUT STRING(ONE_LINE) EDIT(SUBNAME)(A(K));                               00002960
      AUTHOR_ARRAY(LNA)=ONE_LINE;                                               00002970
    END;                                                                        00002980
    VT_ED: LN_COL=0;                                                            00002990
    IF LNA>0 THEN LNA=LNA-1;                                                    00003000
    /*                                                                          00003010
          SET UP STRING FOR - VT  ED  RESUME PG.  LINE                          00003020
    */                                                                          00003030
      PUT STRING(ONE_LINE) EDIT(VT#,ED#,RESUME_PG)(A(10),X(VT_ED_SKIP)          00003040
                ,A(20),X(MARGIN-31-VT_ED_SKIP),A(4));                           00003050
      LN_COL=LN_COL+1;                                                          00003060
      CHAR_COL(LN_COL)=ONE_LINE;                                                00003070
      IF TITLE_LEN<=MARGIN THEN DO;                                             00003080
      PUT STRING(ONE_LINE) EDIT(TITLE)(X(2),A);                                 00003090
      LN_COL=LN_COL+1;                                                          00003100
      CHAR_COL(LN_COL)=ONE_LINE;                                                00003110
```

```
      GO TO IN1;                                                        00003120
                                 END;                                   00003130
      K=MARGIN-2;                                                       00003140
    /*                                                                  00003150
          BREAK UP TITLE ALLOW MAXIMUM OF 14 LINES                      00003160
    */                                                                  00003170
      DO J=K, 2*K-I,J+K-I,J+K-I,J+K-I,J+K-I,J+K-I,J+K-I,J+K-I,J+K-I,J+K-I,J+K-I,00003180
         ,J+K-I,J+K-I,J+K-I;                                            00003190
          IF TITLE_LEN<=J THEN DO;                                      00003200
     PUT STRING(ONE_LINE) EDIT(SUBSTR(TITLE,J-K+1,TITLE_LEN-J+K))(X(2),A);00003210
       LN_COL=LN_COL+1;                                                 00003220
       CHAR_COL(LN_COL)=ONE_LINE;                                       00003230
        GO TO IN1;                                                      00003240
                                 END;                                   00003250
         IF TITLE_LEN<J-K+1 THEN GO TO IN1;                             00003260
        LAB1: DO I=0 TO K-1;                                            00003270
          IF SUBSTR(TITLE,J-I,1)='-'|SUBSTR(TITLE,J-I,1)='-' THEN DO;   00003280
     PUT STRING(ONE_LINE) EDIT(SUBSTR(TITLE,J-K+1,K-I))(X(2),A);        00003290
       LN_COL=LN_COL+1;                                                 00003300
       CHAR_COL(LN_COL)=ONE_LINE;                                       00003310
          GO TO NEXTJ; END LAB1;                                        00003320
         PUT LIST('TITLE ERROR**',TITLE, '**  VT# = ',VT#);             00003330
   NEXTJ:END;                                                           00003340
    IN1: NEW_BLOCK=PAGE_TOTAL+LN_COL+LNA;                               00003350
    /*                                                                  00003360
          FILL IN PRINTER PAGE                                          00003370
    */                                                                  00003380
        IF NEW_BLOCK>LEN_PAGE-INITIAL_LINE THEN DO;                     00003390
          IF COL_COUNT=1 THEN DO I=PAGE_TOTAL + 1 TO LEN_PAGE;          00003400
             PRINTER_PAGE(I)=PRINTER_PAGE(I)||SKIP_LINE;                00003410
                                 END;                                   00003420
                            ELSE DO I=PAGE_TOTAL + 1 TO LEN_PAGE;       00003430
          PRINTER_PAGE(I)=PRINTER_PAGE(I)||COL_SPACE||SKIP_LINE;        00003440
                                 END;                                   00003450
                              COL_COUNT=COL_COUNT+1;                    00003460
                               IF COL_COUNT>#_COLS THEN SIGNAL          00003470
                                    ENDPAGE(OUT);                       00003480
                               IF LNA=0 THEN DO;                        00003490
                                    PAGE_TOTAL=1;                       00003500
                             IF COL_COUNT=1 THEN                        00003510
                                PRINTER_PAGE(1)=PRINTER_PAGE(1)||       00003520
                                    SKIP_LINE;                          00003530
                                ELSE                                    00003540
                              PRINTER_PAGE(1)=PRINTER_PAGE(1)           00003550
                                ||COL_SPACE||SKIP_LINE;                 00003560
                                    END;                                00003570
                                ELSE PAGE_TOTAL=0;                      00003580
                            END;                                        00003590
                      II=PAGE_TOTAL;                                    00003600
                      IF LNA>0 THEN DO I=1 TO LNA;                      00003610
                          II=PAGE_TOTAL+I;                              00003620
          IF COL_COUNT=1 THEN PRINTER_PAGE(II)=PRINTER_PAGE(II)         00003630
                      ||AUTHOR_ARRAY(I); ELSE                           00003640
                    PRINTER_PAGE(II)=PRINTER_PAGE(II)||COL_SPACE||      00003650
                          AUTHOR_ARRAY(I);                              00003660
                                 END;                                   00003670
                      DO I=1 TO LN_COL;                                 00003680
                          II=II+1;                                      00003690
          IF COL_COUNT=1 THEN PRINTER_PAGE(II)=PRINTER_PAGE(II)         00003700
                      ||CHAR_COL(I); ELSE                               00003710
                    PRINTER_PAGE(II)=PRINTER_PAGE(II)||COL_SPACE||      00003720
                          CHAR_COL(I);                                  00003730
```

```
                        END;                                              00003740
                        PAGE_TOTAL=II;                                    00003750
                                                                          00003760
   GO TO IN;                                                              00003770
   FINI: CLOSE FILE(PERSON), FILE(OUT);                                   00003780
   PUT FILE(SYSPRINT) DATA(INDEX_PAGE);                                   00003790
     /*  */                                                              00003800
    /*   END OF LIST ROUTINE  */                                         00003810
     /*  */                                                              00003820
   END;                                                                   00003830
      END FIRST;
```

SUPPORTING SERVICES QUARTERLY

```
/*=============================================================*/00000010
/*                                                             */00000020
/*   Q U A R T E R L Y  S U P P O R T I N G  S E R V I C E S   */00000030
/*              I N D E X   P R O G R A M                      */00000040
/*-------------------------------------------------------------*/00000050
/*                                                             */00000060
/*  PURPOSE:  THE PURPOSE OF THIS PROGRAM IS TO PRODUCE THE    */00000070
/*            QUARTERLY SUPPORTING SERVICES  INDEX  FOR BOTH AIM OR */00000080
/*            ARM.  THE MAIN INPUT IS THE MASTER TAPE DATA SET FOR */00000090
/*            THE QUARTER.  A DISK SUBSET FILE IS CREATED FROM */00000100
/*            MASTER AND THEN THE FILE IS SORTED TO ANOTHER DISK */00000110
/*            FILE USING THE 360 SORT/MERGE.  THE INDEX PRINTOUT */00000120
/*            COMES FROM PRINTING THE SORTED FILE IN A FORMAT */00000130
/*            SPECIFIED BY THE VOCATIONAL CENTER ON AN OUTPUT FILE, */00000140
/*            WHICH IS LATER LISTED ON THE 360/50 FOR UPPER AND */00000150
/*            LOWER CASE PRINTING.  THE SORTED FILE IS SHORTENED */00000160
/*            AND SAVED ON TAPE FOR THE ANNAUL INDEX.          */00000170
/*                                                             */00000180
/*  DEFINITION OF VARIABLES:                                   */00000190
/*     1. AFIELD -- CONVERTED 3-DIGIT SUB-CODE.                */00000200
/*     2. BT -- DESIGNATES BEGIN SHORT TITLE SEGMENT CHARACTER (<). */00000210
/*     3. COL -- NUMBER OF COLUMNS ON AN INDEX PRINTOUT PAGE.  */00000220
/*     4. DUMMY -- STRING USED TO READ AND WRITE FILES.        */00000230
/*     5. ED-NO -- ACCESSION NUMBER.                           */00000240
/*     6. ET -- DESIGNATES END SHORT TITLE SEGMENT CHARACTER (>). */00000250
/*     7. FIELD -- ORIGINAL SUB-CODE FROM MASTER IN 2-DIGIT FORM. */00000260
/*     8. HEAD -- STRING CONTAINING AIM OR ARM, SEASON, AND YEAR. */00000270
/*                IT APPEARS ON THE LINE WITH THE INDEX PAGE NUMBER, */00000280
/*                BUT AT OPPOSITE SIDES OF THE PAGE.           */00000290
/*     9. IHESRTA -- CALL TO SORT/MERGE.                       */00000300
/*    10. LINEH1 -- TITLE PAGE LINE NUMBER ON WHICH PAGE NUMBER AND */00000310
/*                -HEAD- APPEAR.                               */00000320
/*    11. LINEH2 -- SAME AS LINEH1 BUT FOR SUCCEEDING PAGES.   */00000330
/*    12. LINEP1 -- TITLE PAGE LINE NUMBER ON WHICH THE WORD 'PAGE' */00000340
/*                APPEARS ABOVE EACH COLUMN.                   */00000350
/*    13. LINEP2 -- SAME AS LINEP1 BUT FOR SUCCEEDING PAGES.   */00000360
/*    14. LINET -- LINE NUMBER ON ALL PAGES BUT TITLE PAGE ON WHICH */00000370
/*                THE TITLE OF THE INDEX APPEARS.  THERE IS NO TITLE*/00000380
/*                ON THE FIRST PAGE.                           */00000390
/*    15. LENTH -- DATA LENGTHS READ FROM A MASTER RECORD AND USED */00000400
/*                IN THE POSITIONING AND RETRIEVING OF NEEDED DATA. */00000410
/*    16. LH -- LENGTH OF VARIABLE -HEAD-.                     */00000420
/*    17. LTH -- LENGTH OF VARIABLE -TITL-.                    */00000430
/*    18. L_TITLE -- LENGTH OF VARIABLE -TITLE-.               */00000440
/*    19. MARG -- LEFT-HAND MARGIN OF THE INDEX PAGE.          */00000450
/*    20. MASTER -- NAME OF MASTER TAPE.                       */00000460
/*    21. MG -- ARRAY OF COLUMN MARGINS COUNTED FROM COLUMN 1. */00000470
/*    22. NUMB -- DATA LENGTH READ FROM MASTER AND USED IN THE */00000480
/*                POSITIONING OF AND RETRIEVING OF OTHER DATA FROM */00000490
/*                MASTER.                                      */00000500
/*    23. ONE -- SUB-CODE OF '1' REPRESENTING 'TECHNICAL EDUCATION'. */00000510
/*    24. PAGE -- PAGE NUMBER OF INDEX PRINTOUT.               */00000520
/*    25. PAGE_NO -- RESUME PAGE FROM MASTER RECORD.           */00000530
/*    26. PUB_DATE -- PUBLICATION DATE FROM MASTER RECORD.     */00000540
/*    27. QUE -- INDICATES WHETHER THE PAGE NUMBER ON THE TITLE PAGE */00000550
/*                IS IN THE LEFT ('L') OR RIGHT-HAND ('R') MARGIN. */00000560
/*                IT REVERSES ITSELF FOR SUCCEEDING PAGES AND -HEAD- */00000570
/*                IS ALWAYS ON THE OPPOSITE SIDE.              */00000580
/*    28. RETCDE -- CODE RETURNED BY SORT; '0' IS GOOD AND '16' IS */00000590
/*                BAD.                                         */00000600
/*    29. RFIELD -- DEFINES THE INPUT RECORDS FOR SORT.        */00000610
/*    30. SFIELD -- DEFINES WHAT PART(S) OF THE RECORDS WE ARE */00000620
```

```
/*                    SORTING UNDER.                                    */00000630
/*     31. SZ    -- RIGHT-HAND MARGIN FOR INDEX PRINT-OUT PAGE.         */00000640
/*     32. SORTIN -- NAME OF UNSORTED DISK SUBSET FILE.                  */00000650
/*     33. SORTOUT -- NAME OF SORTED DISK SUBSET FILE.                   */00000660
/*     34. SPACE -- NUMBER OF SPACES LEFT BETWEEN VT AND ED NUMBER       */00000670
/*                  WHENEVER THEY ARE PRINTED IN THE INDEX.              */00000680
/*     35. SUM1 -- NUMBER OF POSITIONS ON A MASTER RECORD BETWEEN        */00000690
/*                  RESUME PAGE AND TITLE.                               */00000700
/*     36. SUM2 -- NUMBER OF POSITIONS ON A MASTER RECORD BETWEEN        */00000710
/*                  TITLE AND PUBLICATION DATE.                          */00000720
/*     37. SIZE -- MAXIMUM LENGTH OF INDEX PRINT PAGE.                   */00000730
/*     38. TITL -- TITLE OF THE INDEX.                                   */00000740
/*     39. TITLE -- TITLE OF A DOCUMENT FROM A MASTER RECORD.            */00000750
/*     40. TLE -- ARRAY OF SHORT TITLE SEGMENTS FROM THE TITLE.          */00000760
/*     41. VT_NO -- DOCUMENT NUMBER FROM MASTER.                         */00000770
/*     42. WIDTH -- COLUMN WIDTH IN INDEX PRINTOUT.                      */00000780
/*     43. BLK -- NUMBER OF PIECES OF LENGTH -WIDTH- INTO WHICH A        */00000790
/*                SUB-CODE NAME IS BROKEN.                               */00000800
/*     44. BLKT -- NUMBER OF PIECES OF LENGTH -WIDTH-2- INTO WHICH A     */00000810
/*                 TITLE IS BROKEN.                                      */00000820
/*     45. CODE_NO -- CURRENT SUB-CODE BEING PRINTED IN THE INDEX.       */00000830
/*     46. COL# -- COLUMN SETUP FOR A PAGE IN THE INDEX PRINTOUT.        */00000840
/*                 THE FIRST SUBSCRIPT IS THE COLUMN NUMBER ON THE       */00000850
/*                 PAGE, AND THE SECOND SUBSCRIPT IS THE LINE NUMBER     */00000860
/*                 IN THE COLUMN.                                        */00000870
/*     47. FFIELD -- CONVERTED SUB-CODE ON SUBSET FILE.                  */00000880
/*     48. FLD -- ARRAY OF SUB-CODE NAMES.                               */00000890
/*     49. IMAX -- NUMBER OF COLUMNS PLUS ONE.                           */00000900
/*     50. IMP -- INDICATES START OF NEW SUB-CODE IN INDEX PRINTOUT.     */00000910
/*     51. IND -- COLUMN NUMBER BEING FILLED IN AN INDEX PAGE.           */00000920
/*     52. KNT -- NUMBER OF LINES IN A BLOCK OF DATA.  IT INCLUDES       */00000930
/*                 ONE LINE FOR VT NO., ED NO., AND RESUME PAGE; PLUS    */00000940
/*                 LINES FOR THE TITLE GIVEN BY -BLKT-; PLUS, IF         */00000950
/*                 IMP = 1, LINES FOR THE NEW SUB-CODE NAME GIVEN        */00000960
/*                 BY BLK + 1 (1 IS FOR THE BLANK LINE BETWEEN           */00000970
/*                 DIFFERENT SUB-CODE SECTIONS).                         */00000980
/*     53. LN -- LINE NUMBER ON AN INDEX PAGE BEING FILLED.              */00000990
/*                 (1 < LN < SIZE) THIS IS A CHECK TO MAKE SURE WE DO    */00001000
/*                 NOT EXCEED THE PAGE LENGTH.                           */00001010
/*     54. NN -- LINE NUMBER IN A COLUMN CURRENTLY BEING FILLED.         */00001020
/*                 THIS IS A SUBSCRIPT FOR -COL#-.                       */00001030
/*     55. OUTS -- OUTPUT FILE FOR INDEX PRINTOUT.                       */00001040
/*     56. PC -- PIECES OF BROKEN UP SUB-CODE NAME.                      */00001050
/*     57. PUB -- YEAR PART OF PUBLICATION DATE.                         */00001060
/*     58. SKPH -- COLUMN WHERE -HEAD- IS BEGUN   TO BE PRINTED.         */00001070
/*     59. SKPP -- COLUMN IN WHICH THE PAGE NUMBER PRINTING IS STARTED   */00001080
/*     60. SUB -- NAME OF TAPE WHERE SHORTENED RECORDS ARE SAVED.        */00001090
/*     61. TLP -- PIECES OF BROKEN UP TITLE.                             */00001100
/*                                                                       */00001110
/*  INPUT TO THE PROGRAM:                                                */00001120
/*    MASTER TAPE DATA SET                                               */00001130
/*      DEFINED BY DD CARD IN THE JCL.                                   */00001140
/*      -SYSIN-, INPUT ON CARDS                                          */00001150
/*      THIS INPUT IS ONE CONTINUOUS STREAM ON CARDS.  THE VALUES        */00001160
/*      OF THE VARIABLES LISTED BELOW ARE THE DATA ON THE CARDS.         */00001170
/*      THE ORDER OF THESE VARIABLES IS THEIR ORDER ON THE CARDS         */00001180
/*      AS WELL, AND THE FORMAT LISTED BELOW FOLLOWS WHAT IS ON THE      */00001190
/*      CARDS TOO.  THESE VARIABLES ARE DEFINED IN THE 'DEFINITION'      */00001200
/*      SECTION ABOVE, AND THEIR VALUES CAN BE RECEIVED FROM THE         */00001210
/*      VOCATIONAL CENTER OR THE TYPED WRITE-UP.                         */00001220
/*            -PAGE-      F(4,0)                                         */00001230
/*            -SIZE-      F(4,0)                                         */00001240
```

```
/*          -WIDTH-     F(4,0)                                          */00001250
/*          -MARG-      F(4,0)                                          */00001260
/*          -SZ-        F(4,0)                                          */00001270
/*          -LINEH1-    F(4,0)                                          */00001280
/*          -LINEP1-    F(4,0)                                          */00001290
/*          -LINEH2-    F(4,0)                                          */00001300
/*          -LINET-     F(4,0)                                          */00001310
/*          -LINEP2-    F(4,0)                                          */00001320
/*          -SPACE-     F(4,0)                                          */00001330
/*          -COL-       F(4,0)                                          */00001340
/*          -MG-        (COL) * F(4,0)                                  */00001350
/*          -LTH-       F(4,0)                                          */00001360
/*          -TITL-      A(LTH)                                          */00001370
/*          -LH-        F(4,0)                                          */00001380
/*          -HEAD-      A(LH)                                           */00001390
/*          -QUE-       A(1)                                            */00001400
/*                                                                      */00001410
/*                                                                      */00001420
/*  SPECIAL NOTE:                                                       */00001430
/*     THE OUTPUT FILE SHOULD BE THE PRINTER UNTIL A GOOD INDEX         */00001440
/*     PRINTOUT IS OBTAINED AND THE VOCATIONAL CENTER HAS APPROVED      */00001450
/*     IT.  THEN THE OUTPUT FILE CAN BE CHANGED TO A TAPE AND ONCE      */00001460
/*     THE TAPE IS LISTED ON THE 360/50 WITH THE TN CHAIN, WE HAVE      */00001470
/*     THE UPPER AND LOWER CASE LETTERS IN THE PRINTOUT.                */00001480
/*                                                                      */00001490
/************************************************************************/00001500
                                                                        00001510
VOCIND: PROCEDURE OPTIONS (MAIN);                                       00001520
        DECLARE                                                         00001530
            LINEH1              FIXED BINARY (31,0),                    00001540
            L                                                           00001550
                FIXED BINARY (31,0),                                    00001560
            AFIELD                                                      00001570
                CHARACTER (3),                                          00001580
            BT                  CHARACTER (1) INITIAL ('<'),            00001590
            COL                                                         00001600
                FIXED BINARY (31,0),                                    00001610
            DUMMY               CHARACTER (3736) VARYING,               00001620
            ED_NO               CHARACTER (20),                         00001630
            ET                  CHARACTER (1) INITIAL ('>'),            00001640
            FIELD                                                       00001650
                CHARACTER (2),                                          00001660
            HEAD                CHARACTER (13) VARYING,                 00001670
            I                                                           00001680
                FIXED BINARY (31,0),                                    00001690
            IHESRTA             ENTRY (CHARACTER (45),CHARACTER (33),   00001700
                                FIXED BINARY (31,0),FIXED BINARY (15,0)),00001710
            J                                                           00001720
                FIXED BINARY (31,0),                                    00001730
            LINEH2              FIXED BINARY (31,0),                    00001740
            LINEP1              FIXED BINARY (31,0),                    00001750
            LINEP2              FIXED BINARY (31,0),                    00001760
            LINET               FIXED BINARY (31,0),                    00001770
            LENTH(10)           FIXED BINARY (31,0),                    00001780
            LH                  FIXED BINARY (31,0),                    00001790
            LTH                                                         00001800
                FIXED BINARY (31,0),                                    00001810
            L_TITLE             FIXED BINARY (31,0),                    00001820
            M                                                           00001830
                FIXED BINARY (31,0),                                    00001840
            MARG                FIXED BINARY (31,0),                    00001850
            MASTER                                                      00001860
```

```
        FILE SEQUENTIAL UNBUFFERED INPUT RECORD,                    00001870
    MG(3)               FIXED BINARY (31,0),                        00001880
    N                                                               00001890
        FIXED BINARY (31,0),                                        00001900
    NUMB                FIXED BINARY (31,0),                        00001910
    ONE                                                             00001920
        CHARACTER (1),                                              00001930
    PAGE                                                            00001940
        FIXED BINARY (31,0),                                        00001950
    PAGE_NO             CHARACTER (4),                              00001960
    PUB_DATE                                                        00001970
        CHARACTER (7),                                              00001980
    QUE                 CHARACTER (1),                              00001990
    RETCDE                                                          00002000
        FIXED BINARY (15,0),                                        00002010
    RFIELD              CHARACTER (33),                             00002020
    SFIELD              CHARACTER (45),                             00002030
    SIZE                                                            00002040
        FIXED BINARY (31,0),                                        00002050
    SORTIN                                                          00002060
        FILE SEQUENTIAL RECORD,                                     00002070
    SORTOUT                                                         00002080
        FILE SEQUENTIAL RECORD,                                     00002090
    SPACE               FIXED BINARY (31,0),                        00002100
    SUM1                                                            00002110
        FIXED BINARY (31,0),                                        00002120
    SUM2                                                            00002130
        FIXED BINARY (31,0),                                        00002140
    SZ                  FIXED BINARY (31,0),                        00002150
    TITL                                                            00002160
        CHARACTER (50) VARYING,                                     00002170
    TITLE               CHARACTER (396) VARYING,                    00002180
    TLE(3)                                                          00002190
        CHARACTER (396) VARYING,                                    00002200
    VT_NO                                                           00002210
        CHARACTER (10) VARYING,                                     00002220
    WIDTH                                                           00002230
        FIXED BINARY (31,0),                                        00002240
;                                                                   00002250
                                                                    00002260
OPEN FILE (MASTER),                                                 00002270
    FILE (ERRE) OUTPUT PRINT STREAM LINESIZE (125),                 00002280
    FILE (SORTIN) OUTPUT;                                           00002290
ON ENDFILE (MASTER) GO TO SORT;                                     00002300
                                                                    00002310
                                                                    00002320
/* READ A RECORD FROM THE MASTER TAPE DATA SET.  FIND THE DIVIDER IN 00002330
   THE RECORD.                                                      */00002340
                                                                    00002350
                                                                    00002360
READ_MASTER:                                                        00002370
    READ FILE (MASTER) INTO (DUMMY);                                00002380
    N = INDEX(DUMMY,'||');                                          00002390
    IF N = 0                                                        00002400
      THEN DO;                                                      00002410
        VT_NO = SUBSTR(DUMMY,1,10);                                 00002420
        PUT FILE (ERRE) EDIT                                        00002430
          ('MISSING DIVIDER IN DOCUMENT ',VT_NO)                    00002440
          (SKIP(2),A(28),A(10));                                    00002450
        GO TO READ_MASTER;                                          00002460
        END;                                                        00002470
                                                                    00002480
```

```
                                                                      00002490
     /*  FIND THE NUMBER OF POSITIONS ON THE TAPE BETWEEN THE RESUME PAGE  00002500
         AND THE TITLE.  STORE THIS VALUE IN -SUM1-.  PICK UP THE TITLE   00002510
         LENGTH FROM MASTER.                                              */00002520
-------------------------------------------------------------------------00002530
                                                                          00002540
         SUM1 = 0;                                                        00002550
         GET STRING (DUMMY) EDIT (NUMB,(LENTH(I) DO I = 1 TO NUMB),       00002560
           L_TITLE)  (X(N+1),F(4),(NUMB+1) F(4));                         00002570
         N = N + 9 + (NUMB * 4);                                          00002580
---------IF NUMB¬= 0 THEN DO;                                            00002590
                       LOOP1: DO I = 1 TO NUMB;                           00002600
                              SUM1 = SUM1 + LENTH(I);                     00002610
                              END LOOP1;                                  00002620
                       END;                                               00002630
                                                                          00002640
-------------------------------------------------------------------------00002650
     /*  FIND THE NUMBER OF POSITIONS ON THE TAPE BETWEEN THE TITLE AND   00002660
         THE PUBLICATION DATE.  STORE THIS VALUE IN -SUM2-.               */00002670
                                                                          00002680
                                                                          00002690
         SUM2 = 0;                                                        00002700
-----LOOP2: DO J = 1 TO 2;                                               00002710
         GET STRING (DUMMY) EDIT (NUMB,(LENTH(I) DO I=1 TO NUMB))         00002720
           (X(N),F(4),(NUMB) F(4));                                       00002730
         N = N + 4 + (4 * NUMB);                                          00002740
         IF NUMB ¬= 0 THEN DO;                                            00002750
                        INNER: DO L = 1 TO NUMB;                          00002760
-------------------------------SUM2 = SUM2 + LENTH(L);                   00002770
                               END INNER;                                 00002780
                        END;                                              00002790
         END LOOP2;                                                       00002800
    LOOP3: DO I = 1 TO 8;                                                 00002810
         GET STRING (DUMMY) EDIT (NUMB)                                   00002820
-----------(X(N),F(4));                                                  00002830
         SUM2 = SUM2 + NUMB;                                              00002840
---------N = N + 4;                                                      00002850
         END LOOP3;                                                       00002860
                                                                          00002870
                                                                          00002880
-------/*  NOW WITH THE ABOVE INFORMATION, RETRIEVE THE NECESSARY DATA FROM 00002890
         MASTER.                                                          */00002900
                                                                          00002910
                                                                          00002920
         GET STRING (DUMMY) EDIT (VT_NO,FIELD,ONE,ED_NO,PAGE_NO,TITLE,    00002930
                                  PUB_DATE)                               00002940
-----------(A(10),X(5),A(2),A(1),X(9),A(20),X(7),A(4),X(SUM1),A(L_TITLE)00002950
           ,X(SUM2),A(7));                                                00002960
---------PUT SKIP (4);                                                   00002970
         PUT EDIT (VT_NO,FIELD,ONE,ED_NO,PAGE_NO,TITLE,PUB_DATE) (A);    00002980
                                                                          00002990
                                                                          00003000
-------/*  FORM THE SHORT TITLE FROM THE TAPE TITLE BY CONCATENATING SHORT  00003010
         TITLE SEGMENTS TOGETHER.  A SHORT TITLE SEGMENT STARTS WITH '<'  00003020
         AND ENDS WITH '>'.  THE SHORT TITLE IS USED IN THE INDEX.        */00003030
                                                                          00003040
                                                                          00003050
         I = 0;                                                           00003060
-----RETRN: N = INDEX(TITLE,BT);                                         00003070
         M = INDEX(TITLE,ET);                                             00003080
---------IF (N=0) & (M=0) THEN GO TO LOOP4;                              00003090
         I = I + 1;                                                       00003100
```

```
          TLE(I) = SUBSTR(TITLE,N+1,M-N-1);                                  00003110
          IF SUBSTR(TITLE,M+1,1) ¬= ' ' THEN TLE(I) = TLE(I) ||              00003120
                                              SUBSTR(TITLE,M+1,1);           00003130
          TITLE = SUBSTR(TITLE,M+2);                                         00003140
          GO TO RETRN;                                                       00003150
LOOP4:    DO J = 1 TO I;                                                     00003160
          IF J = 1 THEN TITLE = TLE(1);                                      00003170
                   ELSE TITLE = TITLE || ' ' || TLE(J);                      00003180
          END LOOP4;                                                         00003190
          PUT DATA (TITLE);                                                  00003200
                                                                             00003210
                                                                             00003220
   /*  CONVERT THE TWO-DIGIT SUB-CODE TO THREE DIGITS BY ADDING A ZERO       00003230
       TO THE RIGHT SIDE.  (EX: 45 BECOMES 450).  THEN FORM A RECORD ON      00003240
       THE DISK SUBSET FILE FOR THIS SUB-CODE.                            */ 00003250
                                                                             00003260
                                                                             00003270
          AFIELD = FIELD || '0';                                             00003280
          PUT STRING (DUMMY) EDIT (AFIELD,VT_NO,ED_NO,PAGE_NO,PUB_DATE,      00003290
                                   TITLE)                                    00003300
            (A(3),A(10),A(20),A(4),A(7),A);                                  00003310
          WRITE FILE (SORTIN) FROM (DUMMY);                                  00003320
          PUT DATA (AFIELD);                                                 00003330
                                                                             00003340
                                                                             00003350
   /*  IF THERE IS A '1' FOR 'TECHNICAL EDUCATION' THEN CONVERT THIS         00003360
       SUB-CODE TO '495' AND MAKE A RECORD ON DISK FOR THE SUB-CODE.      */ 00003370
                                                                             00003380
                                                                             00003390
          IF ONE = '1'                                                       00003400
            THEN DO;                                                         00003410
                 AFIELD = '495';                                             00003420
                 PUT STRING (DUMMY) EDIT (AFIELD,VT_NO,ED_NO,PAGE_NO,        00003430
                   PUB_DATE,TITLE)                                           00003440
                   (A(3),A(10),A(20),A(4),A(7),A);                           00003450
                 WRITE FILE (SORTIN) FROM (DUMMY);                           00003460
                 PUT DATA (AFIELD);                                          00003470
                 END;                                                        00003480
          GO TO READ_MASTER;                                                 00003490
                                                                             00003500
                                                                             00003510
   /*  SORT THE DISK SUBSET FILE AND PRINT THE RETURN CODE, AN INDICATION    00003520
       OF A SUCCESSFUL SORT.                                              */ 00003530
                                                                             00003540
                                                                             00003550
SORT:     CLOSE FILE (SORTIN),                                               00003560
                FILE (MASTER);                                               00003570
          SFIELD = ' SORT FIELDS=(5,3,FI,A,11,7,CH,A),SIZE=E9999 ';          00003580
          RFIELD = ' RECORD TYPE=V,LENGTH=(444,,,50) ';                      00003590
          CALL IHESRTA(SFIELD,RFIELD,45056,RETCDE);                          00003600
          PUT FILE (SYSPRINT) EDIT ('RETURN CODE = ',RETCDE)                 00003610
            (SKIP(3),A(14),F(3));                                            00003620
                                                                             00003630
                                                                             00003640
   /*  READ IN THE PAGE SPECIFICATIONS FOR THE INDEX PRINTOUT.            */ 00003650
                                                                             00003660
                                                                             00003670
          GET FILE (SYSIN) EDIT (PAGE,SIZE,WIDTH,MARG,SZ,LINEH1,LINEP1,      00003680
            LINEH2,LINE1,LINEP2,SPACE,COL,(MG(N) DO N = 1 TO COL),LTH,       00003690
            TITL,LH,HEAD,QUE)                                                00003700
            (13 F(4),(COL) F(4),A(LTH),F(4),A(LH),A(1));                     00003710
                                                                             00003720
```

```
/*  THIS BLOCK PRINTS THE INDEX ON THE OUTPUT FILE AND SHORTENS AND
    SAVES THE RECORDS ON TAPE.                                          */

WORK:   BEGIN;
        DECLARE
          BLK
              FIXED BINARY (31,0),
          BLKT
              FIXED BINARY (31,0),
          BREAK               ENTRY(,FIXED BINARY (31,0),,),
          CODE_NO
              FIXED DECIMAL (3,0) INITIAL (999),
          COL#(COL,SIZE-LINEP2-1)  CHARACTER (WIDTH+3),
          FFIELD
              FIXED DECIMAL (3,0),
          FLD(0:13)           CHARACTER (50) VARYING INITIAL
                    ('AGRICULTURAL EDUCATION',
                     'BUSINESS AND OFFICE EDUCATION',
                     'DISTRIBUTIVE EDUCATION',
                     'HEALTH OCCUPATIONS EDUCATION',
                     'HOME ECONOMICS EDUCATION',
                     'INDUSTRIAL ARTS EDUCATION',
                     'MANPOWER ECONOMICS',
                     'OCCUPATIONAL PSYCHOLOGY',
                     'OCCUPATIONAL SOCIOLOGY',
                     '',
                     'TRADE AND INDUSTRIAL EDUCATION',
                     '',
                     'GENERAL VOCATIONAL AND TECHNICAL EDUCATION',
                     'TECHNICAL EDUCATION'),
          IMAX
              FIXED BINARY (31,0),
          IMP                 FIXED BINARY (31,0),
          IND
              FIXED BINARY (31,0),
          K
              FIXED BINARY (31,0),
          KK                  FIXED BINARY (31,0),
          KNT                 FIXED BINARY (31,0),
          LL                  FIXED BINARY (31,0),
          LN
              FIXED BINARY (31,0),
          NN                  FIXED BINARY (31,0),
          OUTS
              FILE,
          PC(14)              CHARACTER (WIDTH) VARYING,
          PUB
              CHARACTER (2),
          SKPH                FIXED BINARY (31,0),
          SKPP                FIXED BINARY (31,0),
          SUB
              FILE SEQUENTIAL RECORD,
          TLP(14)             CHARACTER (WIDTH) VARYING,
        ;

        ON ENDFILE (SORTOUT) GO TO SHORTEN;
        OPEN FILE (SORTOUT),
             FILE (OUTS) OUTPUT STREAM PRINT PAGESIZE(SIZE+4)
        ;
```

```
/*  INITIALIZE VARIABLES FOR PRINTING INDEX.                          */

BLKOT:  DO I = 1 TO SIZE-LINEP2-1;
        DO J = 1 TO COL;
        COL#(J,I) = ' ';
        END BLKOT;
        K = PAGE;
        IND =1;
        IX = COL + 1;
        LN = LINEP1;
        NN = 0;
        IF QUE = 'L' THEN DO;
                          SKPP = MARG;
                          SKPH = SZ - LH + 1;
                          END;
                     ELSE DO;
                          SKPH = MARG;
                          SKPP = SZ - 3;
                          END;

/*  PRINT HEADING ON THE TITLE PAGE.                                  */

        PUT FILE (OUTS) EDIT (PAGE,HEAD)
          (PAGE,LINE(LINEH1),COLUMN(SKPP),F(4),SKIP(0),COLUMN(SKPH),
          A(LH));
        PUT FILE (OUTS) SKIP (LINEP1-LINEH1);
        LP8: DO I = 1 TO COL;
             PUT FILE (OUTS) EDIT ('PAGE')
               (SKIP(0),COLUMN(MARG)+WIDTH-1),A(4));
             END LP8;
        PUT FILE (OUTS) SKIP(1);

/*  READ A RECORD FROM THE SORTED DISK FILE.  BREAK IT UP INTO ITS
    PIECES OF DATA.  BREAK UP THE TITLE INTO PIECES FITTING IN A
    COLUMN OF OUTPUT.                                                 */

READ:   READ FILE (SORTOUT) INTO (DUMMY);
        L = LENGTH(DUMMY);
        GET STRING (DUMMY) EDIT (FFIELD,VT_NO,ED_NO,PAGE_NO,PUB_DATE,
          TITLE)
          (F(3),A(10),A(20),A(4),A(7),A(L-44));
        CALL BREAK (TITLE,WIDTH-2,TLP,BLKT);

/*  IF A NEW SUB-CODE IS BEING STARTED WITH THIS RECORD, FETCH THE
    SUB-CODE NAME AND BREAK IT UP INTO PIECES FITTING IN A COLUMN
    OF OUTPUT.  ASSIGN '1' TO -IMP- TO INDICATE A NEW SUB-CODE.
    ASSIGN -KNT- THE NUMBER OF LINES OF OUTPUT IN THIS BLOCK OF DATA. */

        IF FFIELD¬=CODE_NO
          THEN DO;
               CODE_NO = FFIELD;
               IF FFIELD = 495 THEN L = 13;
                               ELSE L = (FFIELD - 400)/10;
```

```
                IF LENGTH(FLD(L)) > WIDTH                                   00004970
                  THEN DO;                                                  00004980
                       TITLE = FLD(L);                                      00004990
                       CALL BREAK (TITLE,WIDTH,PC,BLK);                     00005000
                       END;                                                 00005010
                  ELSE DO;                                                  00005020
                       BLK = 1;                                             00005030
                       PC(1) = FLD(L);                                      00005040
                       END;                                                 00005050
                KNT = BLK + 2 + BLKT;                                       00005060
                IMP = 1;                                                    00005070
                END;                                                        00005080
                                                                            00005090
                                                                            00005100
    /*  IF THE SAME SUB-CODE IS CONTINUING, SET -IMP- AND -KNT- TO VALUES   00005110
        FOR THIS SITUATION.                                                 */00005120
                                                                            00005130
                                                                            00005140
          ELSE DO;                                                          00005150
               KNT = BLKT + 1;                                              00005160
               IMP = 0;                                                     00005170
               END;                                                         00005180
                                                                            00005190
                                                                            00005200
    /*  IF THIS BLOCK OF DATA CANNOT FIT INTO THE COLUMN BEING FILLED       00005210
        WITHOUT EXCEEDING THE PAGE LENGTH, START A NEW COLUMN.              00005220
                                                                            00005230
                                                                            00005240
          IF (LN + KNT) > SIZE THEN GO TO HEADR;                            00005250
                                                                            00005260
                                                                            00005270
    /*  IF A NEW SUB-CODE IS BEING STARTED, SKIP A LINE IN THE COLUMN AND   00005280
        PRINT THE SUB-CODE NAME.                                            */00005290
                                                                            00005300
                                                                            00005310
  RETURN: IF IMP = 1 THEN DO;                                               00005320
                     NN = NN + 1;                                           00005330
                     LP6: DO I = 1 TO BLK;                                  00005340
                          COL#(IND,NN) = PC(I);                             00005350
                          NN = NN + 1;                                      00005360
                          END LP6;                                          00005370
                     END;                                                   00005380
                                                                            00005390
                                                                            00005400
    /*  PRINT THE VT#, ED#, AND RESUME PAGE IN THE COLUMN, ALL ON ONE       00005410
        LINE.                                                               */00005420
                                                                            00005430
                                                                            00005440
          PUT STRING (COL#(IND,NN)) EDIT (VT_NO,ED_NO,PAGE_NO)              00005450
            (A(10),X(SPACE),A(20),X(WIDTH-30-SPACE-1),A(4));                00005460
          NN = NN + 1;                                                      00005470
                                                                            00005480
                                                                            00005490
    /*  PRINT THE TITLE IN THE COLUMN NEXT.                                 */00005500
                                                                            00005510
                                                                            00005520
  LP7:  DO I = 1 TO BLKT;                                                   00005530
        COL#(IND,NN) = '  ' || TLP(I);                                      00005540
        NN = NN + 1;                                                        00005550
        END LP7;                                                            00005560
        LN = LN + KNT;                                                      00005570
        GO TO READ;                                                         00005580
```

```
                                                                        00005590
                                                                        00005600
     /*  BRANCH HERE WHEN A COLUMN IS AS FULL AS CAN BE.                00005610
         UP THE COLUMN SUBSCRIPT BY ONE.                               */00005620
                                                                        00005630
                                                                        00005640
     HEADR:  IND = IND + 1;                                             00005650
                                                                        00005660
                                                                        00005670
     /*  IF THE LAST COLUMN ON THE PAGE IS FULL, START A NEW PAGE.  FIRST, 00005680
         PRINT OUT THE COLUMNS OF THE FINISHED PAGE.  THEN INCREASE THE 00005690
         INDEX PAGE AND PRINT OUT A TITLE FOR THE NEW PAGE.            */00005700
                                                                        00005710
                                                                        00005720
           IF IND = IMAX                                                00005730
             THEN DO;                                                   00005740
                  IND = 1;                                              00005750
                  HD1: DO LL = 1 TO SIZE-LINEP2-1;                      00005760
                       PUT FILE (OUTS) SKIP(1);                         00005770
                       LP9: DO KK = 1 TO COL;                           00005780
                            PUT FILE (OUTS) EDIT (COL#(KK,LL)           00005790
                             (SKIP(0),COLUMN(MG(KK)),A(WIDTH-3));       00005800
                            COL#(KK,LL) = ' ';                          00005810
                            END LP9;                                    00005820
                       END HD1;                                         00005830
                  PAGE = PAGE + 1;                                      00005840
                  IF SKPP =MARG THEN DO;                                00005850
                                     SKPP = SZ - 3;                     00005860
                                     SKPH = MARG;                       00005870
                                     END;                               00005880
                                ELSE DO;                                00005890
                                     SKPP = MARG;                       00005900
                                     SKPH = SZ - LH + 1;                00005910
                                     END;                               00005920
                  PUT FILE (OUTS) EDIT (PAGE,HEAD,TITL)                 00005930
                    (PAGE,LINE(LINEH2),COLUMN(SKPP),F(4),SKIP(0),COLUMN( 00005940
                    SKPH),A(LH),LINE(LINET),X((SZ-LTH-MARG+1)/2+MARG-1), 00005950
                    A(LTH));                                            00005960
                  PUT FILE (OUTS) SKIP(LINEP2-LINET);                   00005970
                  LP10: DO LL = 1 TO COL;                               00005980
                        PUT FILE (OUTS) EDIT ('PAGE')                   00005990
                          (SKIP(0),COLUMN(MG(LL)+WIDTH-1),A(4));        00006000
                        END LP10;                                       00006010
                  PUT FILE (OUTS) SKIP(1);                              00006020
                  END;                                                  00006030
                                                                        00006040
                                                                        00006050
     /*  REINITIALIZE LINE COUNTERS FOR THE NEW COLUMN.                */00006060
                                                                        00006070
                                                                        00006080
           IF IMP = 1 THEN DO;                                          00006090
                           NN = 0;                                      00006100
                           LN = LINEP2;                                 00006110
                           END;                                         00006120
                      ELSE DO;                                          00006130
                           NN = 1;                                      00006140
                           LN = LINEP2 + 1;                             00006150
                           END;                                         00006160
           IF K = PAGE THEN LN = LN + LINEP1 - LINEP2;                  00006170
           GO TO RETURN;                                                00006180
                                                                        00006190
                                                                        00006200
```

```
      /*  WHEN ALL RECORDS HAVE BEEN PUT INTO COLUMNS, PRINT THE LAST PAGE   00006210
          OF THE INDEX, EVEN THOUGH IT IS NOT COMPLETELY FULL.  WRITE OUT    00006220
          THE LAST PAGE NUMBER OF THE INDEX.                                 */00006230
                                                                             00006240
-------------------------------------------------------------------------------00006250
      SHORTEN:                                                               00006260
              DO I = 1 TO SIZE-LINEP2-1;                                     00006270
              PUT FILE (OUTS) SKIP(1);                                       00006280
              LP11: DO J = 1 TO COL;                                         00006290
                    PUT FILE (OUTS) EDIT (COL(J,I))                          00006300
-----------------------(SKIP(0),COLUMN(MG(J)),A(WIDTH+3));                   00006310
                    END LP11;                                                00006320
              END SHORTEN;                                                   00006330
              PUT FILE (OUTS) PAGE;                                          00006340
              PUT FILE (SYSPRINT) EDIT ('LAST PAGE USED IS ',PAGE)           00006350
                (SKIP(3),A(18),F(4));                                        00006360
-------------CLOSE FILE (OUTS),                                             00006370
                    FILE (SORTOUT);                                          00006380
                                                                             00006390
                                                                             00006400
      /*  SHORTEN THE SORTED RECORDS ON DISK AND PUT THEM ON MAGNETIC TAPE   00006410
          FOR THE ANNUAL INDEX.                                              */00006420
-------------------------------------------------------------------------------00006430
                                                                             00006440
              ON ENDFILE (SORTOUT) GO TO STOP;                               00006450
              OPEN FILE (SORTOUT),                                           00006460
                   FILE (SUB) OUTPUT;                                        00006470
              PUT FILE (SYSPRINT) PAGE;                                      00006480
      RD:     READ FILE (SORTOUT) INTO (DUMMY);                              00006490
              GET STRING (DUMMY) EDIT (AFIELD,VT_NO,PUB)                     00006500
                (A(3),A(10),X(29),A(2));                                     00006510
              VT_NO = SUBSTR(VT_NO,4,3) || SUBSTR(VT_NO,8,3);                00006520
              PUT STRING (DUMMY) EDIT (AFIELD,PUB,VT_NO,'          ')        00006530
                (A(3),A(2),A(6),A(7));                                       00006540
              WRITE FILE (SUB) FROM (DUMMY);                                 00006550
              PUT EDIT (DUMMY) (SKIP(1),A(11));                              00006560
              GO TO RD;                                                      00006570
      STOP:   CLOSE FILE (SUB);                                              00006580
                                                                             00006590
                                                                             00006600
      /*  THIS PROCEDURE BREAKS UP A STRING -BREAKE- INTO PIECES OF LENGTH   00006610
          -LNTH- AND STORES THEM IN -BROKE-.  -NU- IS THE NUMBER OF PIECES   00006620
          IT TOOK TO DO THE JOB.                                             */00006630
                                                                             00006640
-------------------------------------------------------------------------------00006650
      BREAK:  PROCEDURE(BREAKE,LNTH,BROKE,NU);                               00006660
              DECLARE                                                        00006670
                BREAKE              CHARACTER (396) VARYING,                 00006680
                LNTH                FIXED BINARY (31,0),                     00006690
                TEMP                CHARACTER (396) VARYING,                 00006700
                BROKE(14)           CHARACTER (WIDTH) VARYING,               00006710
                NU                  FIXED BINARY (31,0),                     00006720
                L                   FIXED BINARY (31,0),                     00006730
                SUM1                FIXED BINARY (31,0),                     00006740
              ;                                                              00006750
              NU = 0;                                                        00006760
      INIT:   SUM1 = 0;                                                      00006770
              TEMP = BREAKE;                                                 00006780
      SRCH:   L = INDEX(TEMP,'-');                                           00006790
              IF L = 0                                                       00006800
                THEN DO;                                                     00006810
                     L = LENGTH(TEMP);                                       00006820
```

```
                SUM1 = SUM1 + L;                                          00006830
                IF SUM1 > LNTH                                            00006840
                  THEN DO;                                                00006850
                       NU = NU + 1;                                       00006860
                       BROKE(NU) = SUBSTR(BREAKE,1,SUM1-L-1);             00006870
                       NU = NU + 1;                                       00006880
                       BROKE(NU) = SUBSTR(BREAKE,SUM1-L+1);               00006890
                       RETURN;                                            00006900
                       END;                                               00006910
                    ELSE DO;                                              00006920
                       NU = NU + 1;                                       00006930
                       BROKE(NU) = SUBSTR(BREAKE,1,SUM1);                 00006940
                       RETURN;                                            00006950
                       END;                                               00006960
                END;                                                      00006970
              IF L > LNTH                                                 00006980
                THEN DO;                                                  00006990
                PUT FILE (ERRE) EDIT                                      00007000
                  ('TROUBLE IN BREAK WITH ',VT_NO)                        00007010
                  (SKIP(2),A(22),A(10));                                  00007020
                GO TO READ;                                               00007030
                END;                                                      00007040
SUM:          SUM1 = SUM1 + (L - 1);                                      00007050
              IF SUM1 <=LNTH THEN DO;                                     00007060
                                  TEMP = SUBSTR(TEMP,L+1);                00007070
                                  SUM1 = SUM1 + 1;                        00007080
                                  GO TO SRCH;                             00007090
                                  END;                                    00007100
              NU = NU + 1;                                                00007110
              BROKE(NU) = SUBSTR(BREAKE,1,SUM1-L);                        00007120
              BREAKE = SUBSTR(BREAKE,SUM1+2-L);                           00007130
              SUM1 = 0;                                                   00007140
              GO TO SUM;                                                  00007150
              END BREAK;                                                  00007160
                                                                          00007170
              END WORK;                                                   00007180
                                                                          00007190
              CLOSE FILE (ERRE), FILE (SORTOUT);                          00007200
              END VOCIND;                                                 00007210
```

DOCUMENT NUMBER QUARTERLY

```
 DOC_INDEX:   PROCEDURE  OPTIONS (MAIN);                                  00000010
 /*                                                                       00000020
/*  O V E R A L L   P R O G R A M   D E S C R I P T I O N                */00000030
/*                                                                       */00000040
/*                                                                       */00000050
/*    EXTRACT FROM THE MASTER FILE ALL DATA NECESSARY FOR THE            */00000060
/*      DOCUMENT NUMBER INDEX                                            */00000070
/*      TO CREATE THE TITLE TO BE PRINTED                                */00000080
/*    WRITE EACH RECORD ONTO A SORTIN FILE                               */00000090
/*    WHEN THE ENTIRE MASTER FILE HAS BEEN PROCESSED SORT THE SORTIN     */00000100
/*      FILE BY VT NO                                                    */00000110
/*    READ IN THE 2 PRINT CONTROL CARDS WHICH CONTAIN THE DESIRED        */00000120
/*      PRINT SPECIFICATIONS FOR THE DOCUMENT NUMBER INDEX               */00000130
/*  CARD INPUT SPECIFICATIONS - CARD 1                                   */00000140
/*    COL 1 - 3     'AIM' OR 'ARM'                                       */00000150
/*        4 - 6     LINE NO. OF FIRST PRINT LINE - INDX,SEASON,YR,PAGE   */00000160
/*        7 -10     FIRST PAGE NO.                                       */00000170
/*       11-13      LENGTH OF PAGES                                      */00000180
/*       14-16      FIRST PAGE RIGHT OR LEFT - 1=LEFT  2=RIGHT           */00000190
/*       17-19      WIDTH OF MARGIN 1                                    */00000200
/*       20-22      WIDTH OF MARGIN 2 - 0 IF NO MARGIN                   */00000210
/*       23-25      WIDTH OF MARGIN 3 - 0 IF NO MARGIN                   */00000220
/*       26-28      WIDTH OF COLUMN - FROM 'V' IN VT NO TO 'E' IN PAGE   */00000230
/*       29-31      NO. OF COLUMNS                                       */00000240
/*       32-34      NO. OF SPACES BETWEEN VT NO AND ED NO                */00000250
/*       35-37      LINE NO. OF LINE PRINTING 'PAGE' AS COLUMN HEADING   */00000260
/*       38-40      LINE NO. OF FIRST INDEX DETAIL LINE - VT NO ED NO    */00000270
/*       41-43      LINE NO. ON TITLE PAGE OF LINE PRINTING 'PAGE'       */00000280
/*       44-46      LINE NO. ON TITLE PAGE OF FIRST DETAIL LINE          */00000290
/*                                                                       */00000300
/*  CARD INPUT SPECIFICATIONS - CARD 2                                   */00000310
/*    COL 1 - 3     LENGTH OF TITLE                                      */00000320
/*        4 - 6     LINE NO. OF TITLE TO BE PRINTED                      */00000330
/*        7 -80     TITLE ITSELF                                         */00000340
/*    ALLOCATE THE MATRICES USED TO STORE THE INFO TO BE PRINTED         */00000350
/*    READ THE SORTOUT FILE                                              */00000360
/*    EXECUTE THE ROUTINES TO ACCUMULATE THE DATA IN PRINTABLE FORMAT    */00000370
/*      MATRIX LINER - ACCUMULATE 1 PRINT RECORD (IE 1 VT NO REC)        */00000380
/*      MATRIX BUILD - ACCUMULATE AN ENTIRE PAGE IN PRINTABLE FORMAT     */00000390
/*      THE RECORDS ARE STORED IN BUILD IN SEQUENCE BY VT NO BY          */00000400
/*        PRINT COLUMN                                                   */00000410
/*    PRINT AN ENTIRE PAGE FROM MATRIX BUILD WHEN BUILD IS FILLED        */00000420
/*    RE INITIALIZE MATRIX BUILD AND BEGIN ACCUMULATING THE NEXT         */00000430
/*      PAGE IN MATRIX BUILD                                             */00000440
/*    'HANG IT UP' WHEN ALL RECORDS HAVE BEEN PROCESSED + PRINTED        */00000450
/*    ERRONEOUS OUTPUT OR PROGRAM BLOWUPS - MOST LIKELY WILL BE DUE      */00000460
/*      TO INCORRECT PRINT CONTROL CARDS WHICH ARE DEVISED BY THE        */00000470
/*      USER - CHECK THESE CARDS FIRST.                                  */00000480
/*                                                                       */00000490
   DECLARE                                                               00000500
         ANNUAL   FILE OUTPUT RECORD,                                    00000510
         MASTER     FILE INPUT RECORD,                                   00000520
         SORTIN     FILE RECORD,                                         00000530
         SORTOUT    FILE RECORD,                                         00000540
         A_A                    CHAR (3),                               00000550
         ANN                    CHAR (18),                              00000560
         ANNUAL_REC             CHAR (24),                              00000570
         BEL                    FIXED BINARY (31,0) INITIAL (0),        00000580
```

```
BLANK                          CHAR (1) INITIAL (' '),                  00000590
BLD_CODE                       CHAR (1) INITIAL (' '),                  00000600
BT                             CHAR (1) INITIAL ('<'),                  00000610
BUILD   (NOCOL,PGE_LEN)        CONTROLLED  CHAR (COL),                  00000620
BUILDIT                        ENTRY,                                   00000630
COL                            FIXED BINARY (31,0),                     00000640
COL_L                          FIXED BINARY (31,0) INITIAL (0),         00000650
COLCTR                         FIXED BINARY (31,0) INITIAL (1),         00000660
DUM                            CHAR (35),                               00000670
ED_NO                          CHAR (20),                               00000680
EDIT_ISSUE                     CHAR (2),                                00000690
EDIT_VT  (6)                   CHAR (1),                                00000700
ENDTIT                         CHAR (3) INITIAL ('   '),               00000710
ET                             CHAR (1) INITIAL ('>'),                  00000720
FIRST_PAGE                     CHAR (1) INITIAL ('1'),                  00000730
HEAD                           CHAR (74),                               00000740
HINE                           CHAR (118),                              00000750
I                              FIXED BINARY (31,0) INITIAL (0),         00000760
IHESRTA  ENTRY(CHAR(36),CHAR(33),FIXED BINARY(31,0),FIXED               00000770
         BINARY(15,0)),                                                 00000780
INDX                           CHAR (6) VARYING,                        00000790
ISKIP                          FIXED BINARY (31,0),                     00000800
ISSUE                          CHAR (3),                                00000810
J                              FIXED BINARY (31,0) INITIAL (0),         00000820
K                              FIXED BINARY (31,0),                     00000830
L                              FIXED BINARY (31,0),                     00000840
LEN_H                          FIXED BINARY (31,0),                     00000850
LEN_TITLE                      FIXED BINARY (31,0),                     00000860
LENTH (10)                     DECIMAL FIXED (6,0),                     00000870
LIN                            CONTROLLED  CHAR (TIT_L),                00000880
LINE                           CHAR (118) INITIAL ((118) ' '),          00000890
LINE_DETAIL                    FIXED BINARY (31,0),                     00000900
LINE_H                         FIXED BINARY (31,0),                     00000910
LINE_INDX                      FIXED BINARY (31,0),                     00000920
LINE_L                         FIXED BINARY (31,0),                     00000930
LINE_PAGE                      FIXED BINARY (31,0),                     00000940
LINEH                          CHAR (118) INITIAL((118) ' '),           00000950
LINER   (TIT_MAX)              CONTROLLED  CHAR (COL_L),                00000960
LINERF                         CONTROLLED  CHAR (COL),                  00000970
LINESLEFT                      FIXED BINARY (31,0) INITIAL (0),         00000980
LNECTR                         FIXED BINARY (31,0) INITIAL (1),         00000990
M                              FIXED BINARY (31,0),                     00001000
MAST_REC                       CHAR (3736) VARYING,                     00001010
MG1                            FIXED BINARY (31,0),                     00001020
MG2                            FIXED BINARY (31,0),                     00001030
MG3                            FIXED BINARY (31,0),                     00001040
N                              FIXED BINARY (31,0),                     00001050
NOCOL                          FIXED BINARY (31,0),                     00001060
PAGE_NO                        CHAR (4),                                00001070
PGE_LEN                        FIXED BINARY (31,0),                     00001080
PGE_NO                         FIXED BINARY (31,0),                     00001090
PGE_RL                         FIXED BINARY (31,0),                     00001100
PRINTIT                        ENTRY,                                   00001110
RETCDE                         FIXED BINARY (15,0),                     00001120
RFIELD                         CHAR (33),                               00001130
SFIELD                         CHAR (36),                               00001140
```

```
          SPACES                  CHAR (118) INITIAL ((118) ' '),      00001150
          SUBLINER                FIXED BINARY (31,0) INITIAL (1),     00001160
          SUM1                    BINARY FIXED (31,0),                 00001170
          TIT (3)                 CHAR (396) VARYING,                  00001180
          TIT_L                   FIXED BINARY (31,0) INITIAL (0),     00001190
          TIT_MAX                 FIXED BINARY (31,0) INITIAL (0),     00001200
          TITLE                   CHAR (396) VARYING,                  00001210
          TITLE_LEN               DECIMAL FIXED (6,0),                 00001220
          TOP_PAGE                ENTRY,                               00001230
          TP_LIN_DET              FIXED BINARY (31,0),                 00001240
          TP_LIN_PGE              FIXED BINARY (31,0),                 00001250
          UNPACK                  CHAR (11),                           00001260
          VARYER                  DECIMAL FIXED (6,0),                 00001270
          VT_NO                   CHAR (10) VARYING,                   00001280
          YEAR1                   CHAR (2),                            00001290
          VT_ED                   FIXED BINARY (31,0),                 00001300
          YEAR2                   CHAR (2);                            00001310
   BEGIN:                                                              00001320
       OPEN FILE (MASTER),                                             00001330
            FILE (SORTIN) OUTPUT;                                      00001340
   READ_MAST:                                                          00001350
          /*****************************************************/     00001360
          READ FILE (MASTER) INTO (MAST_REC);                          00001370
            ON ENDFILE (MASTER) GO TO SORTIT;                          00001380
          /*****************************************************/     00001390
          N = INDEX(MAST_REC,'||');                                    00001400
          SUM1 = 0;                                                    00001410
          GET STRING (MAST_REC)  EDIT (VARYER) (X(N+1),F(4,0));        00001420
          IF VARYER = 0 THEN GET STRING (MAST_REC) EDIT (TITLE_LEN)    00001430
                                                 (X(N+5),F(4,0));      00001440
            ELSE DO;                                                   00001450
              GET STRING (MAST_REC) EDIT ((LENTH(I) DO I=1 TO VARYER), 00001460
                        TITLE_LEN) (X(N+5),(VARYER+1) F(4,0));         00001470
   THRU_AUTHOR:                                                        00001480
              DO I = 1 TO VARYER;                                      00001490
              SUM1 = SUM1 + LENTH(I);                                  00001500
              END THRU_AUTHOR;                                         00001510
            END;                                                       00001520
          /*****************************************************/     00001530
/*  EXTRACT DATA FROM MASTER FILE - VT NO  ED NO  ISSUE  YEAR        */00001540
/*                                  PAGE NO  TITLE                   */00001550
          GET STRING (MAST_REC) EDIT(VT_NO,ED_NO,ISSUE,YEAR1,YEAR2,    00001560
                  PAGE_NO,TITLE)                                       00001570
          (A(10),X(17),A(20),A(3),A(2),A(2),A(4),X(SUM1),A(TITLE_LEN));00001580
          /*****************************************************/     00001590
          IF TITLE_LEN = 0 THEN DO;                                    00001600
            PUT EDIT ('RECORD WITH A TITLE LENGTH OF ZERO ',VT_NO)     00001610
            (SKIP(1),2 A);                                             00001620
            GO TO READ_MAST;                                           00001630
            END;                                                       00001640
          I = 0;                                                       00001650
/*  ROUTINE TO DETERMINE THE DOCUMENT TITLE BY BREAKING TITLE BY     */00001660
/*     BEGIN AND  END TITLE CHARACTERS                               */00001670
     TIT_BREAK:                                                        00001680
          N = INDEX(TITLE,BT);                                         00001690
          IF N = 0 THEN GO TO CAT;                                     00001700
```

```
            M = INDEX(TITLE,ET);                                                00001710
            SUBSTR (TITLE,M,1) = SUBSTR (TITLE,M+1,1);                          00001720
            L = M - N;                                                          00001730
            I = I + 1;                                                          00001740
            TIT(I) = SUBSTR(TITLE,N+1,L);                                       00001750
            TITLE = SUBSTR(TITLE,M+1);                                          00001760
            GO TO TIT_BREAK;                                                    00001770
      CAT:  TITLE = '';                                                         00001780
      TIT_BREAKER:                                                              00001790
            DO J = 1 TO I;                                                      00001800
            IF J = 1 THEN TITLE = TITLE || TIT(1);                              00001810
              ELSE        TITLE = TITLE || BLANK || TIT(J);                     00001820
            END TIT_BREAKER;                                                    00001830
            TITLE_LEN = LENGTH(TITLE);                                          00001840
   /*  WRITE RECORDS TO BE SORTED BY THE VT NO                                 */00001850
      FOR_SORT:                                                                 00001860
            LEN_TITLE = TITLE_LEN;                                              00001870
            /*******************************************************************/00001880
            PUT STRING (MAST_REC) EDIT (VT_NO,ED_NO,PAGE_NO,LEN_TITLE,          00001890
                        TITLE,ISSUE,YEAR1,YEAR2)                                00001900
                       (A(10),A(20),A(4),F(4),A(TITLE_LEN),A(3),2 A(2));        00001910
            WRITE FILE (SORTIN) FROM (MAST_REC);                                00001920
            PUT EDIT ('UNSORTED EXTRACT ',MAST_REC) (SKIP(2),2 A);              00001930
            /*******************************************************************/00001940
            GO TO READ_MAST;                                                    00001950
            /*******************************************************************/00001960
            /*******************************************************************/00001970
      SORTIT:                                                                   00001980
            CLOSE FILE (SORTIN) , FILE (MASTER);                                00001990
            SFIELD = ' SORT FIELDS=(5,10,CH,A),SIZE=E900 ';                     00002000
            RFIELD = ' RECORD TYPE=V,LENGTH=(450,,,29) ';                       00002010
            CALL IHESRTA(SFIELD,RFIELD,45056,RETCDE);                           00002020
            IF RETCDE = 16 THEN PUT EDIT ('RETURN CODE OF 16') (A(15));         00002030
            /*******************************************************************/00002040
            /*******************************************************************/00002050
            OPEN FILE (SORTOUT);                                                00002060
            ON ENDFILE (SORTOUT) BEGIN;                                         00002070
              PGE_NO = PGE_NO + 1;                                              00002080
              CALL TOP_PAGE;                                                    00002090
              CALL PRINTIT;                                                     00002100
              GO TO PRINT_FIN;                                                  00002110
              END;                                                              00002120
   /*  READ THE 2 PRINT CONTROL CARDS                                          */00002130
      SETUP:                                                                    00002140
            GET FILE (SYSIN) EDIT (A_A,LINE_INDX) (A(3),F(3));                  00002150
            GET FILE (SYSIN) EDIT (PGE_NO,PGE_LEN,PGE_RL,MG1,MG2,               00002160
              MG3,COL,NOCOL,VT_ED,LINE_PAGE,LINE_DETAIL,TP_LIN_PGE,             00002170
              TP_LIN_DET) (F(4),12(F(3)));                                      00002180
            GET FILE (SYSIN) EDIT (DUM) (A(34));                                00002190
            GET FILE (SYSIN)  EDIT (LEN_H,LINE_H,HEAD) (2 F(3),A(74));          00002200
            PGE_NO = PGE_NO - 1;                                                00002210
            COL_L = COL - 3;                                                    00002220
            TIT_L = COL - 5;                                                    00002230
            TIT_MAX = (396/TIT_L) + 4;                                          00002240
            LINE_L = (MG1 + MG2 + MG3 + (NOCOL * COL));                         00002250
            EDIT_ISSUE = SUBSTR(ISSUE,2,2);                                     00002260
```

```
            IF EDIT_ISSUE = 'W ' THEN INDX = 'WINTER';                    00002270
            IF EDIT_ISSUE = 'F ' THEN INDX = 'FALL';                      00002280
            IF EDIT_ISSUE = 'SP' THEN INDX = 'SPRING';                    00002290
            IF EDIT_ISSUE = 'SU' THEN INDX = 'SUMMER';                    00002300
            OPEN FILE (INDIX) OUTPUT STREAM PRINT                         00002310
              PAGESIZE(PGE_LEN);                                          00002320
            PGE_LEN = PGE_LEN - (LINE_DETAIL - 1);                        00002330
            ALLOCATE BUILD;                                               00002340
            ALLOCATE LIN;                                                 00002350
            ALLOCATE LINER;                                               00002360
            ALLOCATE LINERE;                                              00002370
            CALL CLEAN_BUILD;                                             00002380
            PGE_LEN = PGE_LEN - (TP_LIN_DET - LINE_DETAIL);               00002390
            GO TO READ_SORTOUT;                                           00002400
  /* ROUTINE TO PRINT THE HEADINGS AT THE TOP OF PAGE                    */00002410
     TOP_PAGE:  PROCEDURE;                                                00002420
            IF PGE_RL = 1 THEN                                            00002430
              PUT FILE (INDIX) EDIT (PGE_NO,A_A,'/',INDX,' ',YEAR2)       00002440
              (PAGE,LINE(LINE_INDX),X(MG1),F(4),X(LINE_L-(MG1+4+7+        00002450
              LENGTH(INDX))),A(3),A(1),A,A(1),A(2));                      00002460
              ELSE                                                        00002470
            PUT FILE (INDIX) EDIT (A_A,'/',INDX,' ',YEAR2,PGE_NO)         00002480
              (PAGE,LINE(LINE_INDX),X(MG1),A(3),A(1),A,A(1),A(2),         00002490
              X(LINE_L-(MG1+4+7+LENGTH(INDX))),F(4));                     00002500
            IF FIRST_PAGE = '0' THEN                                      00002510
            PUT FILE (INDIX) EDIT (HEAD) (LINE(LINE_H),X((LINE_L/2)-      00002520
              (LEN_H/2)),A(21));                                          00002530
            LINE = SPACES;                                                00002540
            IF NOCOL < 3 THEN GO TO T2;                                   00002550
            SUBSTR(LINE,MG1+MG2+MG3+(3*COL)-3,4) = 'PAGE';                00002560
     T2:    IF NOCOL< 2 THEN GO TO T1;                                    00002570
            SUBSTR(LINE,MG1+MG2+(2*COL)-3,4) = 'PAGE';                    00002580
     T1:    SUBSTR(LINE,MG1+(COL-3),4) = 'PAGE';                          00002590
            HINE = LINE;                                                  00002600
            IF FIRST_PAGE = '0' THEN                                      00002610
            PUT FILE (INDIX) EDIT (HINE) (LINE(LINE_PAGE),A(118));        00002620
             ELSE PUT FILE (INDIX) EDIT (HINE) (LINE(TP_LIN_PGE),A(118));00002630
            IF FIRST_PAGE = '0' THEN                                      00002640
              ISKIP = (LINE_DETAIL - (LINE_PAGE + 1));                    00002650
              ELSE ISKIP = (TP_LIN_DET - (TP_LIN_PGE + 1));               00002660
            PUT FILE (INDIX) EDIT (SPACES) (SKIP(ISKIP),A(118));          00002670
            IF PGE_RL = 1 THEN PGE_RL = 0;  ELSE PGE_RL = 1;              00002680
            RETURN;                                                       00002690
            END TOP_PAGE;                                                 00002700
  /*  READ THE SORTED FILE                                               */00002710
     READ_SORTOUT:                                                        00002720
            /*************************************************************/00002730
            READ FILE (SORTOUT) INTO (MAST_REC);                          00002740
            GET STRING (MAST_REC) EDIT (VT_NO,ED_NO,PAGE_NO,LEN_TITLE,    00002750
                        TITLE,ISSUE,YEAR1,YEAR2)                          00002760
                      (A(10),A(20),A(4),F(4),A(LEN_TITLE),A(3),2 A(2));   00002770
     IF VT_NO='VT 010 625' THEN GO TO READ_SORTOUT;                       00002780
            /*************************************************************/00002790
            CALL ANNUAL_INDEX;                                            00002800
  /*  ROUTINE TO TEMPORARILY STORE THE DOCUMENT TITLE IN PRINTABLE FORM*/00002810
  /*     IN THE MATRIX LINER                                            */00002820
```

```
            LINE = SPACES;                                                    00002830
            PUT STRING (LINE) EDIT (VT_NO,ED_NO,PAGE_NO)                      00002840
              (A(10),X(VT_ED),A(20),X(CCL-(34+VT_ED)),A(4));                  00002850
            SUBLINER = SUBLINER + 1;  LNECTR = LNECTR + 1;                    00002860
            BEG = 1;                                                          00002870
                                                                              00002880
      BRK_TIT:                                                                00002890
            IF LEN_TITLE < TIT_L | LEN_TITLE = TIT_L THEN GO TO END_VT;       00002900
            LIN = SUBSTR(TITLE,BEG,TIT_L);                                    00002910
            IF SUBSTR(TITLE,(BEG+TIT_L),1) = ' ' THEN DO;                     00002920
              LINER(SUBLINER) = ('  ' || LIN);                               00002930
              BEG = BEG + TIT_L + 1;                                          00002940
              LEN_TITLE = LEN_TITLE - (TIT_L + 1);                            00002950
              SUBLINER = SUBLINER + 1;  LNECTR = LNECTR + 1;                  00002960
              GO TO BRK_TIT;                                                  00002970
              END;                                                            00002980
            DO J = TIT_L TO 1 BY -1 WHILE (SUBSTR(LIN,J,1) ¬= ' '); END;      00002990
            LINER(SUBLINER) = ('  ' || SUBSTR(LIN,1,J-1));                   00003000
            BEG = BEG + J;                                                    00003010
            LEN_TITLE = LEN_TITLE - J;                                        00003020
            SUBLINER = SUBLINER + 1;  LNECTR = LNECTR + 1;                    00003030
            GO TO BRK_TIT;                                                    00003040
      END_VT:                                                                 00003050
            LINER(SUBLINER) = ('  ' || SUBSTR(TITLE,BEG,LEN_TITLE));          00003060
            GO TO CHEK_BOT;                                                   00003070
   /*  ROUTINE TO HANDLE BOTTOM OF PAGE IN THE MATRIX BUILD                  */
      CHEK_BOT:                                                               00003080
            LINESLEFT = PGE_LEN - LNECTR;                                     00003090
            IF LINESLEFT > 2 THEN DO;                                         00003100
              BLD_CODE = '1';                                                 00003110
              CALL BUILDIT;                                                   00003120
              BLD_CODE = ' ';                                                 00003130
              GO TO READ_SORTOUT;                                             00003140
              END;                                                            00003150
            IF LINESLEFT >= 0 THEN GO TO FINISH_COL;                          00003160
              IF COLCTR = NOCOL THEN DO;                                      00003170
                LINEH = LINE;                                                 00003180
                PGE_NO = PGE_NO + 1;                                          00003190
                CALL TOP_PAGE;                                                00003200
                LINE = LINEH;                                                 00003210
                CALL PRINTIT;                                                 00003220
                LNECTR = SUBLINER;                                            00003230
                BLD_CODE = '1';                                               00003240
                CALL BUILDIT;                                                 00003250
                BLD_CODE = ' ';                                               00003260
                GO TO READ_SORTOUT;                                           00003270
                END;                                                          00003280
              LNECTR = SUBLINER;                                              00003290
              COLCTR = COLCTR + 1;                                            00003300
              BLD_CODE = '1';                                                 00003310
              CALL BUILDIT;                                                   00003320
              BLD_CODE = ' ';                                                 00003330
              GO TO READ_SORTOUT;                                             00003340
      FINISH_COL:                                                             00003350
            IF COLCTR = NOCOL THEN DO;                                        00003360
              CALL BUILDIT;                                                   00003370
              PGE_NO = PGE_NO + 1;                                            00003380
```

```
            CALL TOP_PAGE;                                                  00003390
            CALL PRINT_IT;                                                  00003400
            GO TO READ_SORTOUT;                                             00003410
            END;                                                            00003420
          CALL BUILDIT;                                                     00003430
          COLCTR = COLCTR + 1;                                              00003440
          LNECTR = 1;                                                       00003450
          GO TO READ_SORTOUT;                                               00003460
          /*****************************************************************/00003470
/*  ROUTINE TO BUILD AN ENTIRE PRINT PAGE IN THE MATRIX BUILD              */00003480
BUILDIT:  PROCEDURE;                                                        00003490
          J = 2;                                                            00003500
          LNECTR = LNECTR + 1;                                              00003510
          I = LNECTR - SUBLINER;                                            00003520
          BUILD(COLCTR,I) = LINE;                                           00003530
          SUBLINER = SUBLINER - 1;                                          00003540
          DO I = SUBLINER  TO 1 BY -1;                                      00003550
          BUILD(COLCTR,LNECTR-I) = LINER (J);                               00003560
          J = J + 1;                                                        00003570
          END;                                                              00003580
          IF BLD_CODE = '1' THEN DO;                                        00003590
            BUILD(COLCTR,LNECTR) = SPACES;                                  00003600
            LNECTR = LNECTR + 1;                                            00003610
            END;                                                            00003620
          DO I = 1 TO TIT_MAX;                                              00003630
          LINER (I) = SPACES;                                               00003640
          END;                                                              00003650
          SUBLINER = 1;                                                     00003660
          RETURN;                                                           00003670
          END BUILDIT;                                                      00003680
          /*****************************************************************/00003690
          /*****************************************************************/00003700
/*  ROUTINE TO PRINT AN ENTIRE PAGE FROM THE  MATRIX BUILD                 */00003710
PRINTIT:      PROCEDURE;                                                    00003720
          DO I = 1 TO PGE_LEN;                                              00003730
          IF NOCOL = 1 THEN PUT FILE (INDIX) EDIT (BUILD(1,I))              00003740
            (SKIP(1),X(MG1),A(COL));                                        00003750
          IF NOCOL = 2 THEN PUT FILE (INDIX) EDIT (BUILD(1,I),              00003760
            BUILD(2,I)) (SKIP(1),X(MG1),A(COL),X(MG2),A(COL));              00003770
          IF NOCOL = 3 THEN PUT FILE (INDIX) EDIT (BUILD(1,I),              00003780
            BUILD(2,I),BUILD(3,I)) (SKIP(1),X(MG1),A(COL),X(MG2),           00003790
            A(COL),X(MG3),A(COL));                                          00003800
          END;                                                              00003810
          CALL CLEAN_BUILD;                                                 00003820
          IF FIRST_PAGE = '1' THEN DO;                                      00003830
            FIRST_PAGE = '0';                                               00003840
            PGE_LEN = PGE_LEN + (TP_LIN_DET - LINE_DETAIL);                 00003850
            END;                                                            00003860
          COLCTR = 1;                                                       00003870
          LNECTR = 1;                                                       00003880
          RETURN;                                                           00003890
          END PRINTIT;                                                      00003900
          /*****************************************************************/00003910
CLEAN_BUILD:  PROCEDURE;                                                    00003920
          DO I = 1 TO PGE_LEN;                                              00003930
          IF NOCOL >= 1 THEN BUILD(1,I) = SPACES;                           00003940
```

```
          IF NOCOL >= 2 THEN BUILD(2,I) = SPACES;                          00003950
          IF NOCOL = 3 THEN BUILD(3,I) = SPACES;                           00003960
            END;                                                           00003970
            RETURN;                                                        00003980
          END CLEAN_BUILD;                                                 00003990
          /*****************************************************************/00004000
/*  ROUTINE TO WRITE OUT THE ANNUAL FILE - VT NO  PAGE NO  ISSUE      */00004010
/*                                         YEAR  -  18 BYTE RECORDS   */00004020
   ANNUAL_INDEX:  PROCEDURE;                                               00004030
          K = 0;                                                           00004040
   EDIT_VTEE:                                                              00004050
          DO J = 1 TO 10;                                                  00004060
          IF SUBSTR(VT_NO,J,1) = 'V' |                                     00004070
             SUBSTR(VT_NO,J,1) = 'T' |                                     00004080
             SUBSTR(VT_NO,J,1) = ' '                                       00004090
             THEN GO TO E_L;                                               00004100
          ELSE DO;                                                         00004110
            K = K + 1;                                                     00004120
            EDIT_VT (K) = SUBSTR(VT_NO,J,1);                               00004130
            END;                                                           00004140
   E_L:   END EDIT_VTEE;                                                   00004150
          /*****************************************************************/00004160
          PUT STRING (ANN) EDIT (EDIT_VT,PAGE_NO,ISSUE,YEAR2,'   ')       00004170
            (6 A(1),A(4),A(3),A(2),A(3));                                  00004180
          WRITE FILE (ANNUAL) FROM (ANN);                                  00004190
          PUT EDIT ('ANNUAL FILE ',ANN) (SKIP(1),2 A);                     00004200
          /*****************************************************************/00004210
          RETURN;                                                          00004220
          END ANNUAL_INDEX;                                                00004230
   PRINT_FIN:                                                              00004240
          CLOSE FILE (SORTOUT) , FILE (INDIX);                             00004250
          END;                                                             00004260
```

CONVERSION OF DOCUMENT NUMBER

```
/*****************************************************************/00000020
/*                                                               */00000030
/*          Q U A R T E R L Y   A N D   A N N U A L              */00000040
/*   C O N V E R S I O N   O F   D O C U M E N T   N U M B E R   */00000050
/*                       I N D E X                               */00000060
/*                                                               */00000070
/*                                                               */00000080
/*  PURPOSE:  THIS PROGRAM CAN GENERATE A QUARTERLY OR AN ANNUAL */00000090
/*            CONVERSION OF DOCUMENT NUMBER INDEX.  IF WE ARE    */00000100
/*            INTERESTED IN A QUARTERLY INDEX, WE INPUT TO THE   */00000110
/*            PROGRAM THE AIM OR ARM CONVERSION TAPE WITH ITS ONE*/00000120
/*            LARGE DATA SET.  FOR AN ANNUAL PRINTOUT, WE USE BOTH*/00000130
/*            AIM AND ARM CONVERSION TAPE DATA SETS (THEY MAY BE ON*/00000140
/*            ONE OR TWO TAPES).  FOR EITHER INDEX, THE VOCATIONAL*/00000150
/*            CENTER SPECIFIES THE PAGE DIMENSIONS AND AN INDEX IS*/00000160
/*            PRODUCED FROM THE TAPE DATA.  THERE IS NO SHORTENING*/00000170
/*            OF RECORDS.  THE QUARTERLY TAPES ARE SAVED AS IS FOR*/00000180
/*            THE ANNUAL INDEX.                                  */00000190
/*                                                               */0000020
/*  DEFINITION OF VARIABLES:                                     */00000210
/*    UNLESS OTHERWISE SPECIFIED, VARIABLES ARE THE SAME FOR EITHER*/00000220
/*    INDEX.                                                     */00000230
/*     1. COL -- NUMBER OF COLUMNS ON AN INDEX PAGE.             */00000240
/*     2. HEAD -- HEADING IN A TOP CORNER OF EVERY PAGE WHICH TELLS*/00000250
/*                THE ISSUE, SEASON, AND YEAR FOR A QUARTERLY    */00000260
/*                PRINTOUT (EX: FALL/AIM 69) OR IT TELLS ISSUE AND*/00000270
/*                YEAR FOR AN ANNUAL PRINTOUT (EX: AIM/ARM 69-70).*/00000280
/*     3. INDX -- INDICATES WHETHER AN ANNUAL OR QUARTERLY INDEX IS*/00000290
/*                TO BE PRODUCED.  'Q' MEANS QUARTERLY; 'A' MEANS*/00000300
/*                ANNUAL.                                        */00000310
/*     4. LINEH1 -- LINE ON THE TITLE PAGE ON WHICH THE INDEX PAGE*/00000320
/*                  NUMBER AND -HEAD- ARE FOUND, AT OPPOSITE ENDS.*/00000330
/*     5. LINEH2 -- SAME AS -LINEH1-BUT FOR ALL PAGES EXCEPT THE  */00000340
/*                  TITLE PAGE.                                  */00000350
/*     6. LINES1 -- LINE ON THE TITLE PAGE OF THE QUARTERLY INDEX*/00000360
/*                  ON WHICH THE SUB-TITLE IS FOUND; LINE ON THE  */00000370
/*                  TITLE PAGE OF THE ANNUAL INDEX RIGHT ABOVE TH*/00000380
/*                  START OF THE ACTUAL INDEX PRINTOUT.           */00000390
/*     7. LINES2 -- SAME AS -LINES1- BUT FOR ALL PAGES EXCEPT THE*/00000400
/*                  TITLE PAGE.                                  */00000410
/*     8. LINET -- LINE ON ALL PAGES EXCEPT THE TITLE PAGE ON WHICH*/00000420
/*     9. LH -- LENGTH OF VARIABLE -HEAD-.                       */00000430
/*    11. LT -- LENGTH OF INDEX TITLE.                           */00000440
/*    12. MARG -- LEFT-HAND MARGIN OF THE INDEX PAGE.            */00000450
/*    13. MG -- ARRAY OF COLUMN MARGINS, EACH MEASURED FROM COLUMN*/00000460
/*              THE TITLE APPEARS.  THERE IS NO TITLE ON THE     */00000470
/*              TITLE PAGE.                                      */00000480
/*            THE ANNUAL INDEX.                                  */00000490
/*    10. LS -- LENGTH OF SUB-TITLE FOR THE QUARTERLY INDEX; N/A FOR*/00000500
/*            1.                                                 */00000510
/*    14. NBH -- NUMBER OF BLOCKS OF PRINT ON THE TITLE PAGE.    */00000520
/*    15. NBR -- NUMBER OF BLOCKS OF PRINT ON ALL OTHER PAGES.   */00000530
/*    16. NPB -- NUMBER OF LINES OF PRINT PER BLOCK.             */00000540
/*    17. PAGE -- PAGE NUMBER OF INDEX PAGES.                    */00000550
/*    18. QUE -- INDICATES WHETHER THE TITLE PAGE WILL BE ON THE */00000560
/*               LEFT ('L') OR RIGHT ('R') SIDE OF THE PUBLICATION.*/00000570
/*    19. SIZE -- NUMBER OF LINES OF PRINT PER PAGE INCLUDING ALL*/00000580
/*                HEADINGS.                                      */00000590
/*    20. SPACE -- USED IN QUARTERLY INDEX ONLY.  NUMBER OF LINES*/00000600
/*                 SKIPPED BETWEEN THE LAST LINE OF PRINT ON THE*/00000610
/*                 TITLE PAGE AND THE FOOTNOTE (SET AT 3 NOW).  */00000620
/*    21. SUB -- SUB-TITLE ON EVERY PAGE OF THE QUARTERLY INDEX. */00000630
```

```
/*     22. SZ -- RIGHT-HAND MARGIN OF THE INDEX PAGE.                          */00000640
/*     23. TITL -- TITLE OF THE INDEX.                                         */00000650
/*     24. WIDTH -- COLUMN WIDTH.                                              */00000660
/*     25. COL# -- A TWO-DIMENSIONAL ARRAY REPRESENTING A PAGE OF              */00000670
/*                 OUTPUT UNDER THE PAGE TITLE.  THE FIRST SUBSCRIPT           */00000680
/*                 REFERENCES A COLUMN NUMBER AND THE SECOND SUBSCRIPT         */00000690
/*                 REFERENCES A LINE IN THE COLUMN.                            */00000700
/*     26. ED -- ACCESSION NUMBER.                                             */00000710
/*     27. IHESRTA -- CALL TO SORT PROGRAM.                                    */00000720
/*     28. IMP -- SIGNIFICANT TO QUARTERLY PROGRAM ONLY.  IMP = 1              */00000730
/*                FOR THE VT TO ED OR MP NUMBER SECTION; IMP = 2 FOR           */00000740
/*                ED OR MP NUMBER TO VT NUMBER SECTION.                        */00000750
/*     29. IND -- COLUMN NUMBER BEING FILLED.                                  */00000760
/*     30. NN -- LINE NUMBER BEING FILLED IN THAT COLUMN.                      */00000770
/*     31. RETCDE -- RETURN CODE RETURNED BY SORT TO INDICATE SUCCESS          */00000780
/*                   OR FAILURE.                                               */00000790
/*     32. RFIELD -- DEFINES THE TYPE OF RECORDS BEING SORTED FOR              */00000800
/*                   SORT.                                                     */00000810
/*     33. SFIELD -- DEFINES THE PARTS OF THE RECORDS TO BE SORTED.            */00000820
/*     34. SKPH -- MARGIN FOR THE HEADING -HEAD- ON ITS PROPER LINE.           */00000830
/*     35. SKPP -- MARGIN FOR THE PAGE NUMBER OF THE INDEX ON ITS              */00000840
/*                 PROPER LINE.                                                */00000850
/*     36. SORTIN -- CONVERSION TAPE DATA SET(S).                              */00000860
/*     37. SORTED -- SORTED DISK SUBSET FILE.                                  */00000870
/*     38. STR -- STRING USED IN INPUTING AND OUTPUTING OF FILE                */00000880
/*                RECORDS.                                                     */00000890
/*     39. UNDER -- USED IN QUARTERLY INDEX ONLY.  THE LINE SEPARATING         */00000900
/*                  THE INDEX AND FOOTNOTE ON THE TITLE PAGE.                  */00000910
/*     40. VT -- DOCUMENT NUMBER FROM MASTER.                                  */00000920
/*                                                                             */00000930
/*   INPUT TO PROGRAM:                                                         */00000940
/*     -CONVERSION TAPE-                                                       */00000950
/*       EITHER THE AIM OR ARM DATA SET IS USED FOR THE QUARTERLY              */00000960
/*       INDEX, OR BOTH DATA SETS ARE USED FOR AN ANNUAL INDEX.  IF            */00000970
/*       THE DATA SETS ARE ON TWO SEPARATE TAPES, CONCATENATE THE              */00000980
/*       TWO DD CARDS.                                                         */00000990
/*     -SYSIN-, INPUT ON CARDS                                                 */00001000
/*       THE CARD INPUT IS ONE CONTINUOUS STREAM OF DATA.  THE VALUES          */00001010
/*       OF THE FOLLOWING VARIABLES ARE FOUND ON THE CARDS IN THE              */00001020
/*       SAME ORDER AS THEY ARE LISTED BELOW AND IN THE FORMAT LISTED          */00001030
/*       BELOW, TOO.                                                           */00001040
/*             -PAGE-              F(4,0)                                      */00001050
/*             -SIZE-              F(4,0)                                      */00001060
/*             -WIDTH-             F(4,0)                                      */00001070
/*             -MARG-              F(4,0)                                      */00001080
/*             -SZ-                F(4,0)                                      */00001090
/*             -LINEH1-            F(4,0)                                      */00001100
/*             -LINES1-            F(4,0)                                      */00001110
/*             -LINEH2-            F(4,0)                                      */00001120
/*             -LINET-             F(4,0)                                      */00001130
/*             -LINES2-            F(4,0)                                      */00001140
/*             -SPACE-             F(4,0)                                      */00001150
/*             -COL-               F(4,0)                                      */00001160
/*             -MG- ARRAY          COL * F(4,0)                                */00001170
/*             -NPB-               F(4,0)                                      */00001180
/*             -NBH-               F(4,0)                                      */00001190
/*             -NBR-               F(4,0)                                      */00001200
/*             -LT-                F(4,0)                                      */00001210
/*             -TITL-              A(LT)                                       */00001220
/*             -LH-                F(4,0)                                      */00001230
/*             -HEAD-              A(LH)                                       */00001240
/*             -INDX-              A(1)                                        */00001250
```

```
/*            -QUE-                    A(1)                                */00001260
/*          FOR QUARTERLY INDEX ONLY, READ IN:                            */00001270
/*            -LS- (FOR SECTION 1) F(4,0)                                 */00001280
/*            -SUB-                    A(LS)                              */00001290
/*            -LS- (FOR SECTION 2) F(4,0)                                 */00001300
/*            -SUB-                    A(LS)                              */00001310
/*      THEY ARE DEFINED IN THE PRECEEDING SECTION AND THEIR VALUES       */00001320
/*      CAN BE FOUND IN TYPED WRITE-UP OR CAN BE SUPPLIED BY THE          */00001330
/*      VOCATIONAL CENTER.                                                */00001340
/*                                                                        */00001350
/*  SPECIAL NOTE:                                                         */00001360
/*    UNTIL THE VOCATIONAL CENTER APPROVES AN INDEX PRINTOUT, USE         */00001370
/*    THE PRINTER AS THE OUTPUT FILE.  THEN AFTER APPROVAL FROM THEM,     */00001380
/*    CHANGE THE OUTPUT FILE TO TAPE.  LIST THE TAPE ON THE 360/50        */00001390
/*    WITH THE TN CHAIN FOR UPPER AND LOWER CASE LETTERS.                 */00001400
/*                                                                        */00001410
/**************************************************************************/00001420
                                                                          00001430
PRNTCV: PROCEDURE OPTIONS (MAIN);                                         00001440
        DECLARE                                                           00001450
          COL                FIXED BINARY (31,0),                         00001460
          HEAD               CHARACTER (13) VARYING,                      00001470
          I                  FIXED BINARY (31,0),                         00001480
          INDX               CHARACTER (1),                               00001490
          LINEH1             FIXED BINARY (31,0),                         00001500
          LINEH2             FIXED BINARY (31,0),                         00001510
          LINES1             FIXED BINARY (31,0),                         00001520
          LINES2             FIXED BINARY (31,0),                         00001530
          LINET              FIXED BINARY (31,0),                         00001540
          LH                 FIXED BINARY (31,0),                         00001550
          LS                 FIXED BINARY (31,0),                         00001560
          LT                 FIXED BINARY (31,0),                         00001570
          M                  FIXED BINARY (31,0),                         00001580
          MARG               FIXED BINARY (31,0),                         00001590
          MG(5)              FIXED BINARY (31,0),                         00001600
          NBH                FIXED BINARY (31,0),                         00001610
          NBR                FIXED BINARY (31,0),                         00001620
          NPB                FIXED BINARY (31,0),                         00001630
          PAGE               FIXED BINARY (31,0),                         00001640
          QUE                CHARACTER (1),                               00001650
          SIZE               FIXED BINARY (31,0),                         00001660
          SPACE              FIXED BINARY (31,0),                         00001670
          SUB                CHARACTER (48),                              00001680
          SZ                 FIXED BINARY (31,0),                         00001690
          TITL               CHARACTER (35),                              00001700
          WIDTH              FIXED BINARY (31,0),                         00001710
        ;                                                                 00001720
                                                                          00001730
                                                                          00001740
    /*  READ IN THE PAGE SPECIFICATIONS FOR THE INDEX PRINTOUT.  ONLY THE 00001750
        QUARTERLY INDEX USES THE SUB-TITLE.                               */00001760
                                                                          00001770
                                                                          00001780
        GET FILE (SYSIN) EDIT (PAGE,SIZE,WIDTH,MARG,SZ,LINEH1,LINES1,    00001790
          LINEH2,LINET,LINES2,SPACE,COL,(MG(I) DO I = 1 TO COL),NPB,      00001800
          NBH,NBR,LT,TITL,LH,HEAD,INDX,QUE)                               00001810
          (16 F(4),(COL) F(4),A(LT),F(4),A(LH),2 A(1));                   00001820
        IF INDX = 'Q' THEN GET FILE (SYSIN) EDIT (LS,SUB) (F(4),A(LS));   00001830
        IF INDX = 'Q' THEN M = LINES2 + 1;                                00001840
                      ELSE M = LINES2;                                    00001850
                                                                          00001860
                                                                          00001870
```

```
/*  PRINTING OF THE INDEX IS ACCOMPLISHED IN THIS BLOCK.                */00001880
                                                                        00001890
                                                                        00001900
WORK:   BEGIN;                                                          00001910
          DECLARE                                                       00001920
          COL#(COL,SIZE-M)  CHARACTER (WIDTH),                          00001930
          ED                CHARACTER (10),                             00001940
          IHESRTA           ENTRY(CHARACTER (45), CHARACTER (27),       00001950
                            FIXED BINARY (31,0), FIXED BINARY (15,0)),  00001960
          INP               FIXED BINARY (31,0),                        00001970
          IND               FIXED BINARY (31,0),                        00001980
          J                 FIXED BINARY (31,0),                        00001990
          K                 FIXED BINARY (31,0),                        00002000
          L                 FIXED BINARY (31,0),                        00002010
          NN                FIXED BINARY (31,0),                        00002020
          RETCDE            FIXED BINARY (15,0),                        00002030
          RFIELD            CHARACTER (27),                             00002040
           SFIELD           CHARACTER (45),                             00002050
          SKPH              FIXED BINARY (31,0),                        00002060
          SKPP              FIXED BINARY (31,0),                        00002070
          SORTIN            FILE SEQUENTIAL RECORD,                     00002080
          SORTOUT           FILE SEQUENTIAL RECORD,                     00002090
          STR               CHARACTER (20),                             00002100
          UNDER             CHARACTER (50) INITIAL ((50)'_'),           00002110
          VT                CHARACTER (10),                             00002120
        ;                                                               00002130
                                                                        00002140
                                                                        00002150
/*  BRANCH HERE WHEN THE SORTED FILE HAS BEEN PRINTED COMPLETELY AND    00002160
    ALL ITS RECORDS READ.                                              */00002170
                                                                        00002180
                                                                        00002190
      ON ENDFILE (SORTOUT) BEGIN;                                       00002200
        CLOSE FILE (SORTOUT);                                           00002210
        INP = INP + 1;                                                  00002220
        IF (NN = 1) & (IND = 1) THEN GO TO TEST;                        00002230
                                                                        00002240
                                                                        00002250
/*  PRINT THE LAST PAGE OF THE INDEX THAT WAS BEING FILLED AND PREPARE  00002260
    FOR A NEW TITLE ON THE NEXT PAGE.                                  */00002270
                                                                        00002280
                                                                        00002290
        LOOP5: DO I = 1 TO L * NPB + L - 1;                             00002300
               PUT FILE (OUTS) SKIP(1);                                 00002310
        LOOP6: DO J = 1 TO COL;                                         00002320
               PUT FILE (OUTS) EDIT (COL#(J,I))                         00002330
                 (SKIP(0),COLUMN(NG(J)),A(WIDTH));                      00002340
               COL#(J,I) = ' ';                                         00002350
               END LOOP6;                                               00002360
               END LOOP5;                                               00002370
        IF SKPP = MARG  THEN DO;                                        00002380
                             SKPP = SZ - 3;                             00002390
                             SKPH = MARG;                               00002400
                             END;                                       00002410
                        ELSE DO;                                        00002420
                             SKPH = SZ - LH + 1;                        00002430
                             SKPP = MARG;                               00002440
                             END;                                       00002450
                                                                        00002460
                                                                        00002470
                        /* THIS IS THE FOOTNOTE THAT APPERS             00002480
                           ON THE TITLE PAGE OF THE QUARTERLY           00002490
```

```
                                INDEX.                                  */
                                                                        00002500
                                                                        00002510
                                                                        00002520
                      IF K=PAGE&INDX='0' THEN  DO;                      00002530
                       K=SIZE-LINEND(OUTS)-SPACE-7;                     00002540
                       PUT FILE(OUTS) SKIP(K);                          00002550
                       K=PAGE;                                          00002560
                        PUT FILE (OUTS) EDIT (UNDER,'1','THIS INDEX IS CUMUL',00002570
                        'ATED ANNUALLY BEGINNING WITH THE FALL ISSUE OF ARM.',00002580
                        '2','ANY DOCUMENT WITH AN MP NUMBER IS AVAILABLE FROM'00002590
                        ,' EDRS BY USING THE','APPROPRIATE MP NUMBER.')  00002600
                        (SKIP(SPACE),COLUMN(MG(1)-3),A(37),SKIP(1),COLUMN(MG  00002610
                        (1)-1),A(1),SKIP(1),COLUMN(MG(1)),A(19),A(51),SKIP(1),00002620
                        COLUMN(MG(1)-1),A(1),SKIP(1),COLUMN(MG(1)),A(48),A(18)00002630
                        ,SKIP(1),COLUMN(MG(1)),A(22));                  00002640
                                        END;                            00002650
              PAGE = PAGE + 1;                                          00002660
    /*  IF WE ARE DOING AN ANNUAL INDEX, THEN AFTER THE VT TO ED NUMBER   00002670
        SECTION IS FINISHED WE END THE PROGRAM.  IF SECTION 2 OF THE      00002680
        QUARTERLY INDEX IS FINISHED, WE END THE PROGRAM ALSO.           */00002690
                                                                        00002700
                                                                        00002710
              TEST: IF IMP ¬= 2 THEN GO TO STOP;                        00002720
              IF INDX = 'A' THEN GO TO STOP;                            00002730
                                                                        00002740
                                                                        00002750
    /*  TO PREPARE FOR SECTION 2 OF THE QUARTERLY INDEX, WE PUT A TITLE   00002760
        ON THE NEXT PAGE AND REVISE THE SORT CONTROL STATEMENTS.  ALSO, A 00002770
        NEW SUB-TITLE IS READ IN.  RETURN FOR FURTHER PROCESSING.       */00002780
                                                                        00002790
                                                                        00002800
              SFIELD = ' SORT FIELDS=(11,2,CH,D,14,7,CH,A),SIZE=E600 '; 00002810
              GET FILE (SYSIN) EDIT (LS,SUB) (F(4),A(LS));              00002820
              PUT FILE (OUTS) EDIT (PAGE,HEAD,TITL,SUB,UNDER)           00002830
                (PAGE,LINE(LINEH2),COLUMN(SKPP),F(4),SKIP(0),COLUMN(SKPH), 00002840
                A(LH),LINE(LINET),X((SZ-MARG+1-LT)/2+MARG-1),A(LT),LINE(  00002850
                LINES2),X((SZ-MARG+1-LS)/2+MARG-1),A(LS),SKIP(0),X((SZ-MARG00002860
                +1-LS)/2+MARG-1),A(LS));                                00002870
              PUT FILE (OUTS) SKIP(1);                                  00002880
              GO TO SORT;                                               00002890
              END;                                                      00002900
                                                                        00002910
                                                                        00002920
    /*  INITIALIZE VARIABLES FOR PRINTING.                            */00002930
                                                                        00002940
                                                                        00002950
            OPEN FILE (OUTS) STREAM OUTPUT PRINT PAGESIZE (SIZE+4)      00002960
            ;                                                           00002970
            IMP = 1;                                                    00002980
            K = PAGE;                                                   00002990
    BLKOT:  DO I = 1 TO SIZE-M;                                         00003000
            DO J = 1 TO COL;                                            00003010
            COL#(J,I) = ' ';                                            00003020
            END BLKOT;                                                  00003030
            IF QUE = 'L' THEN DO;                                       00003040
                              SKPP = MARG;                              00003050
                              SKPH = SZ - LH + 1;                       00003060
                              END;                                      00003070
                         ELSE DO;                                       00003080
                              SKPH = MARG;                              00003090
                              SKPP = SZ - 3;                            00003100
                              END;                                      00003110
```

```
        L = NBH;                                                        00003120
                                                                        00003130
                                                                        00003140
    /*  PUT A TITLE ON THE TITLE PAGE.  PUT A SUB-TITLE ON THE PAGE IF  00003150
        WE ARE DOING A QUARTERLY INDEX.                                 */00003160
                                                                        00003170
                                                                        00003180
        PUT FILE (OUTS) EDIT (PAGE,HEAD)                                00003190
          (PAGE,LINE(LINEH1),COLUMN(SKPP),F(4),SKIP(0),COLUMN(SKPH),    00003200
          A(LH));                                                       00003210
        IF INDX = 'A' THEN DO;                                          00003220
                    PUT FILE(OUTS) SKIP(LINES1-LINEH1);                 00003230
                            GO TO DEF;                                  00003240
                            END;                                        00003250
        PUT FILE (OUTS) EDIT ('2',SUB,UNDER)                            00003260
          (LINE(LINES1-1),X((SZ-MARG+1-LS)/2+MARG-1+LS),A(1),LINE(      00003270
          LINES1),X((SZ-MARG+1-LS)/2+MARG-1),A(LS),SKIP(0),X((SZ-MARG+100003280
          -LS)/2+MARG-1),A(LS));                                        00003290
        PUT FILE (OUTS) SKIP (1);                                       00003300
                                                                        00003310
                                                                        00003320
    /*  MAKE PREPARATIONS FOR SORT AND CALL THE SORT ROUTINE.  PRINT THE 00003330
        RETURN CODE.                                                    */00003340
                                                                        00003350
                                                                        00003360
    DEF:    SFIELD = ' SORT FIELDS=(4,7,CH,A),SIZE=E600 ';              00003370
            RFIELD = ' RECORD TYPE=F,LENGTH=(20) ';                     00003380
    SORT:   CALL IHESRTA(SFIELD,RFIELD,45056,RETCDE);                   00003390
            PUT EDIT ('RETURN CODE IS ',RETCDE)                         00003400
              (SKIP(3),A(15),F(3));                                     00003410
                                                                        00003420
                                                                        00003430
    /*  PUT DATA INTO A COLUMN UNTIL THE COLUMN IS FULL.                */00003440
                                                                        00003450
                                                                        00003460
            NN = 1;                                                     00003470
            IND = 1;                                                    00003480
            OPEN FILE (SORTOUT);                                        00003490
    LOOP1:  DO I = 1 TO L;                                              00003500
    LOOP2:  DO J = 1 TO NPB;                                            00003510
            READ FILE (SORTOUT) INTO (STR);                             00003520
            GET STRING (STR) EDIT (VT,ED) (A(10),A(10));                00003530
            IF IMP = 1 THEN PUT STRING (COL#(IND,NN)) EDIT (VT,ED)      00003540
                              (A(10),X(WIDTH-20),A(10));                00003550
                       ELSE PUT STRING (COL#(IND,NN)) EDIT (ED,VT)      00003560
                              (A(10),X(WIDTH-20),A(10));                00003570
            NN = NN + 1;                                                00003580
            END LOOP2;                                                  00003590
            NN = NN + 1;                                                00003600
            END LOOP1;                                                  00003610
                                                                        00003620
                                                                        00003630
    /*  UP THE COLUMN SUBSCRIPT SO A NEW COLUMN CAN BE FILLED.  IF ALL  00003640
        THE COLUMNS, HOWEVER, HAVE BEEN FILLED FOR A PAGE, WE MUST PUT  00003650
        THE PAGE ON THE OUTPUT FILE FIRST.  PRINT A NEW TITLE ON THE    00003660
        NEXT PAGE AND INITIALIZE AS NECESSARY.  THEN FILL A NEW COLUMN. */00003670
                                                                        00003680
                                                                        00003690
            IND = IND + 1;                                              00003700
            IF IND = (COL+1)                                            00003710
              THEN DO;                                                  00003720
                   IND = 1;                                             00003730
```

```
LOOP3: DO I = 1 TO L * NPB + L - 1;                                    00003740
       PUT FILE (OUTS) SKIP (1);                                       00003750
LOOP4: DO J = 1 TO COL;                                                00003760
       PUT FILE (OUTS) EDIT (COL#(J,I))                                00003770
         (SKIP(0),COLUMN(MG(J)),A(WIDTH));                             00003780
       COL#(J,I) = ' ';                                                00003790
       END LOOP4;                                                      00003800
       END LOOP3;                                                      00003810
IF INDX = 'A' THEN GO TO NO_                                           00003820
                                                                       00003830
                                                                       00003840
                 /* THIS IS THE FOOTNOTE THAT APPERS                  00003850
                    ON THE TITLE PAGE OF THE QUARTERLY                 00003860
                    INDEX.                                  */         00003870
                                                                       00003880
                                                                       00003890
IF K = PAGE THEN                                                       00003900
  PUT FILE (OUTS) EDIT (UNDER,'1','THIS INDEX IS CUMUL',               00003910
  'ATED ANNUALLY BEGINNING WITH THE FALL ISSUE OF ARM.',               00003920
  '2','ANY DOCUMENT WITH AN NP NUMBER IS AVAILABLE FROM'               00003930
  ,' EDRS BY USING THE','APPROPRIATE NP NUMBER.')                      00003940
  (SKIP(SPACE),COLUMN(MG(1)-3),A(37),SKIP(1),COLUMN(MG                 00003950
  (1)-1),A(1),SKIP(1),COLUMN(MG(1)),A(19),A(51),SKIP(1),               00003960
  COLUMN(MG(1)-1),A(1),SKIP(1),COLUMN(MG(1)),A(48),A(18)               00003970
  ,SKIP(1),COLUMN(MG(1)),A(22));                                       00003980
NO_NR:  L = NBR;                                                       00003990
PAGE = PAGE + 1;                                                       00004000
IF SKPP = MARG  THEN DO;                                               00004010
                     SKPP = SZ - 3;                                    00004020
                     SKPH = MARG;                                      00004030
                     END;                                              00004040
                ELSE DO;                                               00004050
                     SKPH = SZ - LH + 1;                               00004060
                     SKPP = MARG;                                      00004070
                     END;                                              00004080
PUT FILE (OUTS) EDIT (PAGE,HEAD,TITL)                                  00004090
  (PAGE,LINE(LINEH2),COLUMN(SKPP),F(4),SKIP(0),COLUMN(                 00004100
  SKPH),A(LH),LINE(LINET),X((SZ-MARG+1-LT)/2+MARG-1),A(                00004110
  LT));                                                                00004120
IF INDX = 'A' THEN DO;                                                 00004130
                   PUT FILE (OUTS) SKIP(LINES2-LINET);                 00004140
                   NN = 1;                                             00004150
                   GO TO LOOP1;                                        00004160
                   END;                                                00004170
PUT FILE (OUTS) EDIT (SUB,UNDER)                                       00004180
  (LINE(LINES2),X((SZ-MARG+1-LS)/2+MARG-1),A(LS),SKIP(0)               00004190
  ,X((SZ-MARG+1-LS)/2+MARG-1),A(LS));                                  00004200
PUT FILE (OUTS) SKIP(1);                                               00004210
END;                                                                   00004220
NN = 1;                                                                00004230
GO TO LOOP1;                                                           00004240
                                                                       00004250
/*  CLOSE ALL FILES AND WRITE OUT THE LAST PAGE NUMBER USED IN THE     00004260
    INDEX PRINTOUT.                                                    00004270
                                                                */     00004280
                                                                       00004290
STOP:  PUT FILE (OUTS) PAGE;                                           00004300
       CLOSE FILE (OUTS);                                              00004310
       PAGE = PAGE - 1;                                                00004320
       PUT FILE (SYSPRINT) EDIT ('LAST PAGE USED IS ',PAGE)            00004330
         (SKIP(3),A(18),F(4));                                         00004340
                                                                       00004350
```

```
END WORK;                                                    00004360
                                                             00004370
END PRNTCV;                                                  00004380
```

SUBJECT ANNUAL

```
/*                                                                      */
/*                                                                      */00000020
/*     A N N U A L   S U B J E C T   I N D E X   P R O G R A M          */00000030
/*                                                                      */00000040
/*  PROGRAMMER:  RONNIE ROSEN                                           */00000050
/*                                                                      */00000060
/*  PURPOSE:  THE PROGRAM PRODUCES THE ANNUAL SUBJECT INDEX.  IT        */00000070
/*            TAKES THE 8 DATA SETS FROM THE QUARTERLY OUTPUT AND       */00000080
/*            CONCATENATES THEM TO FORM ONE LARGE DATA SET.  THE        */00000090
/*            SORT/MERGE SORTS THESE RECORDS AND WE PRINT THEM UNDER    */00000100
/*            SPECIFIED FORMAT TO FORM THE INDEX PRINTOUT.  THE         */00000110
/*            SORTED DATA IS DELETED, BUT THE 8 DATA SETS WHICH ARE     */00000120
/*            ON 2 TAPES, ARE KEPT.                                     */00000130
/*                                                                      */00000140
/*  DEFINITION OF VARIABLES:                                            */00000150
/*     1. BLK -- NUMBER OF PIECES INTO WHICH THE SUBJECT KEYWORD        */00000160
/*               MUST BE BROKEN UP SO THAT IT WILL FIT IN A COLUMN      */00000170
/*               WIDTH.                                                 */00000180
/*     2. CNT -- THE NUMBER OF DOCUMENTS WRI      ABOUT SOME KEYWORD    */00000190
/*               IN THE YEAR -P-.                                       */00000200
/*     3. COL -- NUMBER OF COLUMNS IN A PRINT-OUT PAGE.                 */00000210
/*     4. DUMMY -- INPUT OR OUTPUT STRING USED TO CREATE OR RETRIEVE    */00000220
/*                 RECORDS.                                             */00000230
/*     5. FV -- THE VT NUMBERS OF THE DOCUMENTS WRITTEN ABOUT SOME      */00000240
/*              KEYWORD  IN THE YEAR -P-.                               */00000250
/*     6. FRST -- THE PAGE NUMBER OF THE TITLE PAGE.                    */00000260
/*     7. FVT -- A SINGLE VT NUMBER.                                    */00000270
/*     8. HEAD -- PART OF HEADING OF EVERY PAGE OF THE INDEX WHICH      */00000280
/*                LOOKS LIKE 'AIM/ARM 69-70'.                           */00000290
/*     9. IHESRTA -- PL/1 CALL TO SORT.                                 */00000300
/*    10. IMAX -- INDICATES WHEN ALL COLUMNS OF A PAGE ARE FULL.        */00000310
/*    11. IMP -- INDICATES ALL DATA HAS BEEN PUT INTO INDEX FORM        */00000320
/*               AND THE SORTED FILE HAS BEEN READ ALL THROUGH.         */00000330
/*    12. IND -- REFERENCES TO CURRENT COLUMN BEING FILLED.             */00000340
/*    13. KNT -- THE NUMBER OF LINES OF OUTPUT IT TAKES TO PRINT       */00000350
/*               PUBLICATION YEAR AND THE VT NUMBERS OF DOCUMENTS      */00000360
/*               WRITTEN THEN. THE VT NUMBERS ARE BROKEN UP INTO       */00000370
/*               THOUSAND INTERVALS; 0 - 999 LISTED FIRST; 1000-1999   */00000380
/*               LISTED NEXT, ETC.  EACH INTERVAL STARTS A NEW LINE.   */00000390
/*    14. LIM -- THE NUMBER OF VT NUMBERS THAT COULD FIT ON ONE LINE    */00000400
/*               OF A COLUMN.                                           */00000410
/*    15. LINEH1 -- THE LINE NUMBER ON THE TITLE PAGE ON WHICH THE      */00000420
/*                  PAGE NUMBER AND -HEAD- APPEAR.                      */00000430
/*    16. LINEH2 -- SAME AS -LINEH1- EXCEPT FOR ALL SUCCEEDING PAGES.   */00000440
/*    17. LINES1 -- LINE NUMBER ON THE TITLE PAGE DIRECTLY ABOVE THE    */00000450
/*                  STARTING LINE OF THE INDEX PRINTOUT.                */00000460
/*    18. LINES2 -- SAME AS -LINES1- EXCEPT FOR ALL SUCCEEDING PAGES.   */00000470
/*    19. LINET -- LINE NUMBER ON ALL PAGES EXCEPT THE TITLE PAGE       */00000480
/*                 WHERE THE TITLE APPEARS.                             */00000490
/*    20. LN -- COUNTER WHICH TELLS THE LAST LINE ON A PAGE THAT        */00000500
/*              WAS FILLED WITH INFORMATION.  IT COUNTS HEADING AND     */00000510
/*              INDEX, AND MAKES SURE WE DON'T EXCEDE THE MAXIMUM       */00000520
/*              PAGE SIZE.                                              */00000530
/*    21. LNE -- COUNTS THE NUMBER OF VT NUMBERS BEING PUT ON A LINE    */00000540
/*               SO THAT THE LIMIT IS NOT EXCEDED.                      */00000550
/*    22. LTH -- LENGTH OF THE INDEX TITLE.                             */00000560
/*    23. MARG -- LEFT-HAND MARGIN OF THE INDEX PAGE.                   */00000570
/*    24. MG -- ARRAY OF COLU    ARGINS EACH MEASURED FROM COLUMN 1.    */00000580
/*    25. MT -- DESIGNATES WHI   NTERVAL THE VT NUMBER FALLS IN.        */00000590
/*              MT * 1000 TO MT * 1000 + 999, WHERE MT = 0, 1, 2,...    */00000600
/*              IS THE INTERVAL.                                        */00000610
/*    26. NN -- COUNTS THE LINES AS A COLUMN IS FILLED.  THIS DOES      */00000620
```

```
/*              NOT INCLUDE HEADING.                                          */00000630
/*     27. OUTS -- THE OUTPUT FILE FOR THE INDEX PRINTOUT.                    */00000640
/*     28. P -- ARRAY OF PUBLICATION DATES FOR A KEYWORD  NAME WHEN           */00000650
/*              DOCUMENTS WERE PUBLISHED ABOUT THAT KEYWORD NAME.             */00000660
/*     29. PAGE -- PAGE NUMBER OF AN INDEX PAGE.                              */00000670
/*     30. PREV -- KEYWORD  WHICH HAS JUST HAD ITS DATA ADDED TO THE          */00000680
/*                 INDEX PRINTOUT.                                            */00000690
/*     31. PUB -- A PUBLICATION YEAR FROM A SUBSET FILE RECORD.               */00000700
/*     32. QUE -- INDICATES WHETHER THE TITLE PAGE WILL BE ON THE             */00000710
/*                LEFT (L) OR RIGHT (R) SIDE OF THE PUBLICATION.              */00000720
/*     33. RETCDE -- THE CODE RETURNED BY SORT; '0' MEANS SUCCESS AND         */00000730
/*                   '16' MEANS FAILURE.                                      */00000740
/*     34. RFIELD -- DEFINES THE RECORDS BEING SORTED FOR SORT.               */00000750
/*     35. SFIELD -- DEFINES WHAT PARTS OF THE RECORDS ARE USED AS            */00000760
/*                   THE SORT KEY FOR SORT.                                   */00000770
/*     36. SIZE -- THE MAXIMUM LENGTH OF A PAGE, INDEX AND HEADING.           */00000780
/*     37. SKPH -- THE COLUMN ON -LINEH1- OR -LINEH2- WHERE -HEAD-            */00000790
/*                 SHOULD BEGIN TO BE PRINTED.                                */00000800
/*     38. SKPP -- THE COLUMN ON -LINEH1- OR -LINEH2- WHERE -PAGE-            */00000810
/*                 SHOULD BEGIN TO BE PRINTED.                                */00000820
/*     39. SORTIN -- THE EIGHT DATA SETS ON TAPE THAT ARE THE INPUT           */00000830
/*                   TO SORT.                                                 */00000840
/*     40. SORTOUT -- OUTPUT FILE WITH THE SORTED RECORDS.                    */00000850
/*     41. SUBJ -- KEYWORD NAME FROM A SUBSET FILE RECORD.                    */00000860
/*     42. SZ -- RIGHT-HAND MARGIN OF A PAGE.                                 */00000870
/*     43. TITLE -- INDEX TITLE.                                              */00000880
/*     44. V -- CHARACTER ARRAY OF VT NUMBERS FOR DOCUMENTS WRITTEN           */00000890
/*              ABOUT SOME KEYWORD IN SOME YEAR.                              */00000900
/*     45. VT -- DOCUMENT NUMBER FROM SOME SUBSET FILE RECORD.                */00000910
/*     46. WD -- NUMBER OF MAXIMUM SIGNIFICANT DIGITS IN A VT NUMBER.         */00000920
/*     47. WIDTH -- COLUMN WIDTH IN A PRINTOUT.                               */00000930
/*     48. COL# -- REPRESENTS THE INDEX PART OF A PAGE.                       */00000940
/*     49. PC -- PIECES INTO WHICH A KEYWORD IS BROKEN UP SO THAT IT          */00000950
/*               FITS INTO A COLUMN.                                          */00000960
/*     50. TEMP-OUT -- THE ACTUAL SET-UP OF HOW THE COMPUTER PREPARES         */00000970
/*                     THE DATA FOR OUTPUT.  THE DATA IS A DATE AND           */00000980
/*                     ALL ITS VT NUMBERS.                                    */00000990
/*                                                                            */00001000
/*  INPUT TO THE PROGRAM:                                                     */00001010
/*    -SORTIN-                                                                */00001020
/*      THIS IS YOUR 8 DATA SETS SAVED FROM THE QUARTERLY RUNS,               */00001030
/*      CONCATENATED TOGETHER TO FORM ONE LARGE DATA SET TO BE                */00001040
/*      PROCESSED.                                                            */00001050
/*    -SYSIN-, INPU  ON CARDS                                                 */00001060
/*      THIS IS A CONTINUOUS STREAM OF DATA ON SEVERAL CARDS.  THE            */00001070
/*      DATA IS VALUES FOR THE VARIABLES LISTED BELOW, IN THE SAME            */00001080
/*      ORDER AS BELOW.                                                       */00001090
/*           -PAGE-          F(4)                                             */00001100
/*           -SIZE-          F(4)                                             */00001110
/*           -WIDTH-         F(4)                                             */00001120
/*           -WD-            F(4)                                             */00001130
/*           -MARG-          F(4)                                             */00001140
/*           -SZ-            F(4)                                             */00001150
/*           -LINEH1-        F(4)                                             */00001160
/*           -LINES1-        F(4)                                             */00001170
/*           -LINEH2-        F(4)                                             */00001180
/*           -LINET-         F(4)                                             */00001190
/*           -LINES2-        F(4)                                             */00001200
/*           -COL-           F(4)                                             */00001210
/*           -MG- ARRAY      COL * F(4)                                       */00001220
/*           -LTH-           F(4)                                             */00001230
/*           -TITLE-         A(LTH)                                           */00001240
```

```
/*          -HEAD-          A(13)                                          */00001250
/*          -OUE-           A(1)                                           */00001260
/*      THESE VARIABLES GIVE THE OUTPUT SPECIFICATIONS AND ARE             */00001270
/*      DEFINED INDIVIDUALLY IN THE PRECEDING SECTION.  THEIR VALUES       */00001280
/*      CAN BE OBTAINED FROM THE TYPED WRITE-UP AND THE VOCATIONAL         */00001290
/*      CENTER PERSONNEL.                                                  */00001300
/*                                                                         */00001310
/*  SPECIAL NOTE:                                                          */00001320
/*    PRINT THE ANNUAL INDEX ON PLAIN PAPER UNTIL IT HAS BEEN              */00001330
/*    APPROVED BY THE VOCATIONAL CENTER.  THEN IT CAN BE PRINTED ON        */00001340
/*    MAGNETIC TAPE AND THE TAPE LISTED ON THE 360/50 WITH THE TN          */00001350
/*    CHAIN FOR AN UPPER AND LOWER CASE INDEX PRINTOUT.                    */00001360
/*                                                                         */00001370
/***************************************************************************/00001380
                                                                           00001390
ANNSUB:  PROCEDURE OPTIONS (MAIN);                                         00001400
         DECLARE                                                           00001410
           BLK                 FIXED BINARY (31,0),                        00001420
           CNT(15)             FIXED BINARY (31,0),                        00001430
           COL                 FIXED BINARY (31,0),                        00001440
           DUMMY               CHARACTER (208),                            00001450
           FV(15,75)           FIXED BINARY (31,0) INITIAL ((1125)0),      00001460
           FRST                FIXED BINARY (31,0),                        00001470
           FVT                 FIXED BINARY (31,0),                        00001480
           HEAD                CHARACTER (13),                             00001490
           I                   FIXED BINARY (31,0),                        00001500
           IHESRTA             ENTRY(CHARACTER(59),CHARACTER(28),          00001510
                                 FIXED BINARY (31,0),FIXED BINARY          00001520
                                 (15,0)),                                  00001530
           IMAX                FIXED BINARY (31,0),                        00001540
           INP                 FIXED BINARY (31,0),                        00001550
           IND                 FIXED BINARY (31,0),                        00001560
           J                   FIXED BINARY (31,0),                        00001570
           K                   FIXED BINARY (31,0),                        00001580
           KNT                 FIXED BINARY (31,0),                        00001590
           L                   FIXED BINARY (31,0),                        00001600
           LIM                 FIXED BINARY (31,0),                        00001610
           LINEH1              FIXED BINARY (31,0),                        00001620
           LINEH2              FIXED BINARY (31,0),                        00001630
           LINES1              FIXED BINARY (31,0),                        00001640
           LINES2              FIXED BINARY (31,0),                        00001650
           LINET               FIXED BINARY (31,0),                        00001660
           LN                  FIXED BINARY (31,0),                        00001670
           LNE                 FIXED BINARY (31,0),                        00001680
           LTH                 FIXED BINARY (31,0),                        00001690
           MARG                FIXED BINARY (31,0),                        00001700
           MG(3)               FIXED BINARY (31,0),                        00001710
           MT                  FIXED BINARY (31,0),                        00001720
           N                   FIXED BINARY (31,0),                        00001730
           NN                  FIXED BINARY (31,0),                        00001740
           OUTS                FILE STREAM,                                00001750
           P(15)               CHARACTER (2),                              00001760
           PAGE                FIXED BINARY (31,0),                        00001770
           PREV                CHARACTER (132) VARYING,                    00001780
           PUB                 CHARACTER (2),                              00001790
           QUE                 CHARACTER (1),                              00001800
           RETCDE              FIXED BINARY (15,0),                        00001810
           RFIELD              CHARACTER (28),                             00001820
           SFIELD              CHARACTER (59),                             00001830
           SIZE                FIXED BINARY (31,0),                        00001840
           SKPH                FIXED BINARY (31,0),                        00001850
           SKPP                FIXED BINARY (31,0),                        00001860
```

```
          SORTIN              FILE SEQUENTIAL RECORD,                    00001870
          SORTOUT             FILE SEQUENTIAL RECORD,                    00001880
          SUBJ                CHARACTER (200) VARYING,                   00001890
          SZ                  FIXED BINARY (31,0),                       00001900
          TITLE               CHARACTER (50) VARYING,                    00001910
          V(15,75)            CHARACTER (6) VARYING,                     00001920
          VT                  CHARACTER (6) VARYING,                     00001930
          WD                  FIXED BINARY (31,0),                       00001940
          WIDTH               FIXED BINARY (31,0),                       00001950
        ;                                                                00001960
                                                                         00001970
                                                                         00001980
/*  SORT THE 8 DATA SETS, FIRST BY SUBJECT NAME, THEN BY PUBLICATION*/00001990
/*  DATE, AND FINALLY BY VT NUMBER, ALL IN ASCENDING ORDER.  PRINT  */00002000
/*  THE RETURN CODE, AN INDICATION OF SUCCESS.  WRITE OUT THE SORTED*/00002010
/*  FILE AS A CHECK IN CASE OF TROUBLE.                             */00002020
                                                                         00002030
                                                                         00002040
SFIELD=' SORT FIELDS=(1,200,CH,A,201,2,CH,A,203,6,CH,A),SIZE=E9999 ';    00002050
 RFIELD=' RECORD TYPE=F,LENGTH=(208) ';                                  00002060
CALL IHESRTA (SFIELD,RFIELD,45056,RETCDE);                               00002070
PUT FILE (SYSPRINT) EDIT (RETCDE)  (SKIP(3),F(3));                       00002080
                                                                         00002090
                                                                         00002100
/*  READ IN THE OUTPUT SPECIFICATIONS FOR THE INDEX PRINTOUT.       */00002110
                                                                         00002120
                                                                         00002130
GET FILE (SYSIN) EDIT (PAGE,SIZE,WIDTH,WD,MARG,SZ,LINEH1,LINES1000021400
  ,LINEH2,LINET,LINES2,COL,(MG(I) DO I = 1 TO COL),LTH,TITLE,            00002150
  HEAD,QUE)                                                              00002160
  (13 F(4),(COL) F(4),A(LTH),A(13),A(1));                                00002170
                                                                         00002180
                                                                         00002190
/*  PRINTING IS DONE IN THIS BLOCK.                                 */00002200
                                                                         00002210
                                                                         00002220
BUSINESS:                                                                00002230
        BEGIN;                                                           00002240
        DECLARE                                                          00002250
          COL#(COL,SIZE-LINES2)  CHARACTER (WIDTH),                      00002260
          HEADR                  ENTRY (FIXED BINARY (31,0)),            00002270
          PC(4)                  CHARACTER (WIDTH) VARYING,              00002280
          TEMP_OUT(SIZE-LINES2)  CHARACTER (WIDTH) VARYING,              00002290
        ;                                                                00002300
                                                                         00002310
                                                                         00002320
/*  WHEN THE FILE OF SORTED RECORDS HAS BEEN FULLY READ AND PUT INTO*/00002330
/*  THE INDEX, WE TRANSFER HERE.  THE ASSIGNMENTS MADE HERE WILL    */00002340
/*  ALLOW TRANSFER BACK TO THE FLOW OF ACTION, WHERE THE SET-UP FOR */00002350
/*  THE LAST PAGE CAN BE PRINTED, AND WILL CAUSE PROGRAM            */00002360
/*  TERMINATION.                                                    */00002370
                                                                         00002380
                                                                         00002390
        ON ENDFILE (SORTOUT) BEGIN;                                      00002400
          IMP = 1;                                                       00002410
          SUBJ = 'ZZ';                                                   00002420
          GO TO TEST;                                                    00002430
          END;                                                           00002440
                                                                         00002450
                                                                         00002460
/*  MAKE INITIALIZATIONS FOR PRINTING OF THE INDEX.                 */00002470
                                                                         00002480
```

```
                                                                        00002490
BLKOT:  DO I = 1 TO SIZE-LINES2;                                        00002500
        DO J = 1 TO COL;                                                00002510
        COL#(J,I) = ' ';                                                00002520
        END;                                                            00002530
        END BLKOT;                                                      00002540
        OPEN FILE (SORTOUT),                                            00002550
             FILE (OUTS) OUTPUT STREAM PRINT PAGESIZE(SIZE + 4)         00002560
        ;                                                               00002570
        IMP = 0;                                                        00002580
        IND = 1;                                                        00002590
        FRST = PAGE;                                                    00002600
        IMAX = COL + 1;                                                 00002610
        LIM = (WIDTH - 4)/(WD + 2);                                     00002620
        PUT DATA (LIM);                                                 00002630
        NN = 0;                                                         00002640
        LN = 9;                                                         00002650
        IF QUE = 'L' THEN DO;                                           00002660
                          SKPP = MARG;                                  00002670
                          SKPH = SZ - 12;                               00002680
                          END;                                          00002690
                     ELSE DO;                                           00002700
                          SKPP = SZ - 3;                                00002710
                          SKPH = MARG;                                  00002720
                          END;                                          00002730
                                                                        00002740
/*   PRINT THE HEADING FOR THE TITLE PAGE OF THE INDEX ON THE OUTPUT    */00002750
/*   FILE.                                                              */00002760
                                                                        */00002770
                                                                        00002780
        PUT FILE (OUTS) EDIT (PAGE,HEAD)                                00002790
          (PAGE,LINE(LINEH1),COLUMN(SKPP),F(4),SKIP(0),COLUMN(SKPH),    00002800
          A(13));                                                       00002810
        PUT FILE (OUTS) SKIP (LINES1-LINEH1);                           00002820
                                                                        00002830
                                                                        00002840
                                                                        00002850
/*   READ IN THE VERY FIRST SORTED RECORD AND BREAK THE RECORD UP       */00002860
/*   INTO ITS INDIVIDUAL PIECES OF DATA.                                */00002870
                                                                        00002880
                                                                        00002890
READ:   READ FILE (SORTOUT) INTO (DUMMY);                               00002900
        GET STRING (DUMMY) EDIT (SUBJ,PUB,FVT)                         00002910
             (A(200),A(2),F(6,0));                                      00002920
        GET STRING (DUMMY) EDIT (VT)                                    00002930
             (X(202),A(6));                                             00002940
        CALL DELET;                                                     00002950
        N = INDEX(SUBJ,'"');                                            00002960
        IF N = 0 THEN PUT EDIT ('***NO END OF SUBJECT NAME IN ',SUBJ,   00002970
                       ' IN DOCUMENT ',VT,'***')                        00002980
                      (SKIP(2),A(29),A(200),A(13),A(6),A(13));          00002990
        SUBJ = SUBSTR(SUBJ,1,N-1);                                      00003000
                                                                        00003010
                                                                        00003020
/*   EVERY TIME A NEW KEYWORD'S DATA IS STARTED, WE BRANCH HERE.        */00003030
/*   INITIALIZATIONS ARE MADE IN THE ARRAYS THAT STORE ALL THE DATA     */00003040
/*   FOR THE KEYWORD.  ARRAY -P- WILL HOLD THE PUBLICATION DATES,       */00003050
/*   -CNT- WILL TELL HOW MANY DOCUMENTS WERE PUBLISHED UNDER EACH       */00003060
/*   DATE, -V- AND -FV- WILL HOLD ALL DOCUMENT NUMBERS FOR EACH         */00003070
/*   DATE, AND -I- WILL HOLD THE NUMBER OF DATES IN -P-,  -PREV- IS     */00003080
/*   THE KEYWORD NAME.                                                  */00003090
                                                                        00003100
```

```
                                                                        00003110
INITL:  I = 1;                                                          00003120
        J = 1;                                                          00003130
        PREV = SUBJ;                                                    00003140
        IF PUB = '  ' THEN PUB = 'ND';                                  00003150
        CNT(1) = 1;                                                     00003160
        P(1) = PUB;                                                     00003170
        V(1,1) = VT;                                                    00003180
        FV(1,1) = FVT;                                                  00003190
                                                                        00003200
                                                                        00003210
/*   READ IN A RECORD FROM THE SORTED FILE AND BREAK IT UP INTO ITS     */00003220
/*   INDIVIDUAL DATA.                                                   */00003230
                                                                        00003240
                                                                        00003250
RD_ON:  READ FILE (SORTOUT) INTO (DUMMY);                               00003260
        GET STRING (DUMMY) EDIT (SUBJ,PUB,FVT)                         00003270
             (A(200),A(2),F(6,0));                                      00003280
        GET STRING (DUMMY) EDIT (VT)                                    00003290
             (X(202),A(6));                                             00003300
        CALL DELET;                                                     00003310
        IF PUB = '  ' THEN PUB = 'ND';                                 00003320
        N = INDEX(SUBJ,'''');                                           00003330
        IF N = 0 THEN PUT EDIT ('***NO END OF SUBJECT NAME IN ',SUBJ,   00003340
                        ' IN DOCUMENT ',VT,'***')                       00003350
                       (SKIP(2),A(29),A(200),A(13),A(6),A(13));         00003360
        SUBJ = SUBSTR(SUBJ,1,N-1);                                      00003370
                                                                        00003380
                                                                        00003390
/*   IF THIS NEW RECORD'S DATA BELONGS WITH THE KEYWORD WE ARE          */00003400
/*   PROCESSING, ADD THE DATA TO THE ARRAYS.                            */00003410
                                                                        00003420
                                                                        00003430
TEST:   IF SUBJ = PREV                                                  00003440
          THEN DO;                                                      00003450
                                                                        00003460
                                                                        00003470
            /* IF THE PUBLICATION DATE IS THE SAME AS THE              */00003480
            /* PREVIOUS RECORD, ADD ANOTHER VT NUMBE                   */00003490
            /* THAT DATE.                                              */00003500
                                                                        00003510
                                                                        00003520
              IF P(I) = PUB THEN DO;                                    00003530
                                 J = J + 1;                             00003540
                                 CNT(I) = J;                            00003550
                                 V(I,J) = VT;                           00003560
                                 FV(I,J) = FVT;                         00003570
                                 END;                                   00003580
                                                                        00003590
                                                                        00003600
            /* IF THE PUBLICATION DATE IS DIFFERENT, START             */00003610
            /* A PLACE FOR A NEW DATE WITH THIS VT NUMBER              */00003620
            /* AS THE FIRST NUMBER UNDER IT.                           */00003630
                                                                        00003640
                                                                        00003650
                               ELSE DO;                                 00003660
                                 I = I + 1;                             00003670
                                 P(I) = PUB;                            00003680
                                 J = 1;                                 00003690
                                 V(I,1) = VT;                           00003700
                                 FV(I,J) = FVT;                         00003710
                                 CNT(I) = 1;                            00003720
```

```
                        END;                                                00003730
                                                                            00003740
                                                                            00003750
          /* GO BACK AND READ THE NEXT RECORD.              */             00003760
                                                                            00003770
                                                                            00003780
               GO TO RD_ON;                                                 00003790
               END;                                                         00003800
                                                                            00003810
                                                                            00003820
   /*   IF THIS NEW RECORD'S DATA STARTS THE NEXT KEYWORD, WE MUST ADD  */00003830
   /*   THE OLD KEYWORD'S DATA TO THE INDEX PAGE BEING FILLED.  THEN    */00003840
   /*   WE WILL START THE SAME PROCEDURE OVER AGAIN FOR THE NEW KEYWORD.*/00003850
                                                                            00003860
                                                                            00003870
          ELSE DO;                                                          00003880
               IF I>15 THEN PUT EDIT ('***MORE THAN 15 DATES UNDER-- ',     00003890
                  PREV,' ***')(SKIP(2),A(30),A(200),A(4));                  00003900
                                                                            00003910
                                                                            00003920
               /* BREAK THE KEYWORD NAME INTO SMALLER PIECES IF */          00003930
               /* IT EXCEDES THE COLUMN WIDTH IN LENGTH.        */          00003940
                                                                            00003950
                                                                            00003960
               L = LENGTH(PREV);                                            00003970
               IF L>WIDTH THEN CALL BREAK(PREV,WIDTH,PC,BLK);               00003980
                          ELSE DO;                                          00003990
                               BLK = 1;                                     00004000
                               PC(1) = PREV;                                00004010
                               END;                                         00004020
                                                                            00004030
                                                                            00004040
               /* FOR EACH PUBLICATION DATE, WE WILL ADD THE    */          00004050
               /* DATE AND ALL VT NUMBERS TO THE INDEX PAGE, AND*/          00004060
               /* THEN DO THE SAME FOR THE NEXT DATE.           */          00004070
                                                                            00004080
                                                                            00004090
               LOOP1: DO J=1 TO I;                                          00004100
                      IF CNT(J)>75 THEN PUT EDIT                            00004110
                        ('***MORE THAN  75 DOCUMENTS UNDER ',P(J),          00004120
                        ' FOR KEYWORD ',PC(1),' ***')                       00004130
                        (SKIP(2),A(33),A(2),A(13),A(WIDTH),A(4));           00004140
                      KNT = 1;                                              00004150
                      TEMP_OUT(1) = P(J) || ': ';                           00004160
                      LNE = 0;                                              00004170
                      MT = FV(J,1)/1000;                                    00004180
                                                                            00004190
                                                                            00004200
                    /* FIRST THE DATE AND VT NUMBERS,    */                 00004210
                    /* BROKEN UP INTO THOUSAND INTERVALS,*/                 00004220
                    /* ARE PUT INTO A TEMPORARY ARRAY    */                 00004230
                    /* (TEMP_OUT).  THEIR APPEARANCE IN  */                 00004240
                    /* THIS ARRAY IS IDENTICAL TO THEIR  */                 00004250
                    /* APPEARANCE IN THE INDEX.  IT TOOK */                 00004260
                    /* -KNT- LINES IN  TEMP_OUT- TO GET  */                 00004270
                    /* THIS BLOCK OF DATA STORED.        */                 00004280
                                                                            00004290
                                                                            00004300
                      LOOP2: DO L = 1 TO CNT(J);                            00004310
                             IF MT¬=(FV(J,L)/1000) THEN DO;                 00004320
                                                  MT=FV(J,L)/1000;          00004330
                                                  LNE=0;                    00004340
```

```
                                        KNT=KNT+1;                      00004350
                                        TEMP_OUT(KNT)=                  00004360
                                                '     ';                00004370
                                        END;                            00004380
                    IF (L¬=1) & (LNE¬=0) THEN TEMP_OUT(KNT) =           00004390
                      TEMP_OUT(KNT) || ',';                             00004400
                    TEMP_OUT(KNT)=TEMP_OUT(KNT)||' '||V(J,L);           00004410
                    LNE = LNE + 1;                                      00004420
                    IF (LNE = LIM)                                      00004430
                     &(FV(J,L+1)/1000=MT)                               00004440
                     &(L ¬= CNT(J))                                     00004450
                                THEN DO;                                00004460
                                     TEMP_OUT(KNT)=TEMP_OUT(KNT00004470
                                       ) || ',';                        00004480
                                     KNT=KNT+1;                         00004490
                                     LNE = 0;                           00004500
                                     TEMP_OUT(KNT)='     ';             00004510
                                     END;                               00004520
                    END LOOP2;                                          00004530
                                                                        00004540
                                                                        00004550
/* IF THIS IS THE FIRST DATE FOR THIS*/                                 00004560
/* KEYWORD, WRITE THE KEYWORD NAME IN*/                                 00004570
/* THE COLUMN BEING FILLED.          */                                00004580
                                                                        00004590
                                                                        00004600
                    IF J ¬= 1 THEN GO TO LP2;                           00004610
                    IF (LN+BLK+2) > SIZE THEN CALL HEADR(0);            00004620
                    NN = NN + 1;                                        00004630
                    LN = LN + 1;                                        00004640
                    LP1: DO L = 1 TO BLK;                               00004650
                         COL#(IND,NN) = PC(L);                          00004660
                         NN = NN + 1;                                   00004670
                         END LP1;                                       00004680
                    LN = LN + BLK;                                      00004690
                                                                        00004700
                                                                        00004710
/* START TO ADD THE -TEMP_OUT- BLOCK */                                 00004720
/* TO THE COLUMN.  START NEW COLUMNS */                                 00004730
/* AS NECESSARY, BUT REPEAT THE      */                                00004740
/* KEYWORD NAME AND THE DATE WHEN    */                                00004750
/* THIS HAPPENS.  WHEN THE COMPLETE  */                                 00004760
/* BLOCK HAS BEEN PUT IN COLUMNS,    */                                00004770
/* REPEAT THE COMPLETE PROCEDURE FOR */                                 00004780
/* THE NEXT DATE.                    */                                00004790
                                                                        00004800
                                                                        00004810
                    LP2: DO L = 1 TO KNT;                               00004820
                         IF (LN + 1) > SIZE                             00004830
                           THEN DO;                                     00004840
                                CALL HEADR(1);                          00004850
                                DO N = 1 TO BLK;                        00004860
                                COL#(IND,NN) = PC(N);                   00004870
                                NN = NN + 1;                            00004880
                                LN = LN + 1;                            00004890
                                END;                                    00004900
                                IF LENGTH(PC(BLK))+12 > WIDTH           00004910
                                  THEN DO;                              00004920
                                       COL#(IND,NN)='(CONTINUED)';      00004930
                                       NN = NN + 1;                     00004940
                                       LN = LN + 1;                     00004950
                                       END;                             00004960
```

```
                                    ELSE COL#(IND,NN-1)= PC(BLK) ||          00004970
                                           ' (CONTINUED)';                   00004980
                                  IF L ^= 1 THEN TEMP_OUT(L)= P(J) ||        00004990
                                    ': ' || SUBSTR(TEMP_OUT(L),5);           00005000
                                  END;                                       00005010
                             COL#(IND,NN) = TEMP_OUT(L);                     00005020
                             NN = NN + 1;                                    00005030
                             LN = LN + 1;                                    00005040
                             END LP2;                                        00005050
                        END LOOP1;                                           00005060
                                                                             00005070
                                                                             00005080
          /* WHEN ALL THE KEYWORD'S INFORMATION HAS BEEN    */              00005090
          /* ADDED TO COLUMNS OF INDEX PRINTOUT, WE CAN     */              00005100
          /* GO BACK AND REPEAT ALL THIS FOR THE NEXT       */              00005110
          /* KEYWORD.  IF THERE ARE NO MORE KEYWORDS, DO    */              00005120
          /* NOT BRANCH BACK.                               */              00005130
                                                                             00005140
                                                                             00005150
              IF IMP = 0 THEN GO TO INITL;                                   00005160
              END;                                                           00005170
                                                                             00005180
                                                                             00005190
/*   ADD THE LAST PAGE BEING FILLED TO THE OUTPUT FILE.  TERMINATE    */00005200
/*   THE PROGRAM.                                                     */00005210
                                                                             00005220
                                                                             00005230
LOOP4:  DO K = 1 TO SIZE-LINES2;                                             00005240
        PUT FILE (OUTS) SKIP(1);                                             00005250
LOOP5:  DO L = 1 TO COL;                                                     00005260
        PUT FILE (OUTS) EDIT (COL#(L,K))                                     00005270
          (SKIP(0),COLUMN(MG(L)),A(WIDTH));                                  00005280
        END LOOP5;                                                           00005290
        END LOOP4;                                                           00005300
STOP:   PUT FILE (OUTS) PAGE;                                                00005310
        PUT DATA (PAGE);                                                     00005320
        CLOSE FILE (SORTOUT),                                                00005330
              FILE (OUTS);                                                   00005340
                                                                             00005350
                                                                             00005360
/*   THIS PROCEDURE IS CALLED WHEN A NEW COLUMN IS STARTED.  WHEN ALL*/00005370
/*   COLUMNS ON A PAGE HAVE BEEN FILLED, WE PUT THE COLUMNS ON THE   */00005380
/*   OUTPUT FILE.  A NEW HEADING IS PUT ON THE FILE  FOR THE NEW     */00005390
/*   PAGE.  COLUMN INITIALIZATION PROCEDES AS FOR ANY NEW COLUMN.    */00005400
                                                                             00005410
                                                                             00005420
HEADR:  PROCEDURE (NP);                                                      00005430
        DECLARE                                                              00005440
          (K,L)              FIXED BINARY (31,0),                            00005450
          NP                 FIXED BINARY (31,0),                            00005460
        ;                                                                    00005470
        IND = IND + 1;                                                       00005480
        IF IND = IMAX                                                        00005490
          THEN DO;                                                           00005500
               IND = 1;                                                      00005510
               HD1: DO L = 1 TO SIZE-LINES2;                                 00005520
                    PUT FILE (OUTS) SKIP(1);                                 00005530
               HD2: DO K = 1 TO COL;                                         00005540
                    PUT FILE (OUTS) EDIT (COL#(K,L))                         00005550
                      (SKIP(0),COLUMN(MG(K)),A(WIDTH));                      00005560
                    COL#(K,L) = ' ';                                         00005570
                    END HD2;                                                 00005580
```

```
                END HD1;                                                    00005590
              PAGE = PAGE + 1;                                              00005600
              IF SKPP = MARG THEN DO;                                       00005610
                                  SKPP = SZ - 3;                            00005620
                                  SKPH = MARG;                              00005630
                                  END;                                      00005640
                             ELSE DO;                                       00005650
                                  SKPP = MARG;                              00005660
                                  SKPH = SZ - 12;                           00005670
                                  END;                                      00005680
              PUT FILE (OUTS) EDIT (PAGE,HEAD,TITLE)                        00005690
                (PAGE,LINE(LINEH2),COLUMN(SKPP),F(4),SKIP(0),COLUMN(        00005700
                SKPH),A(13),LINE(LINET),X((SZ-LTH-MARG+1)/2+MARG-1),        00005710
                A(LTH));                                                    00005720
              PUT FILE (OUTS) SKIP(LINES2-LINET);                           00005730
              END;                                                          00005740
         IF FRST = PAGE THEN LN = LINES1;                                   00005750
                        ELSE LN = LINES2;                                   00005760
         NN = NP;                                                           00005770
         IF NN = 0 THEN LN = LN - 1;                                        00005780
         RETURN;                                                            00005790
         END HEADR;                                                         00005800
                                                                            00005810
                                                                            00005820
/*   THIS PROCEDURE IS CALLED TO BREAK UP A STRING (BREAKE) INTO      */00005830
/*   PIECES OF LENGTH -LNTH-.  THE PIECES ARE STORED IN -PC- AND      */00005840
/*   THERE ARE -NU- OF THEM.                                          */00005850
                                                                            00005860
                                                                            00005870
BREAK:   PROCEDURE(BREAKE,LNTH,BROKE,NU);                                   00005880
         DECLARE                                                            00005890
           BREAKE              CHARACTER(200) VARYING,                      00005900
           LNTH                FIXED BINARY (31,0),                         00005910
           BROKE(4)            CHARACTER (WIDTH) VARYING,                   00005920
           NU                  FIXED BINARY (31,0),                         00005930
           L                   FIXED BINARY (31,0),                         00005940
           SUM1                FIXED BINARY (31,0),                         00005950
           TEMP                CHARACTER (132) VARYING,                     00005960
         ;                                                                  00005970
         NU = 0;                                                            00005980
INIT:    SUM1 = 0;                                                          00005990
         TEMP = BREAKE;                                                     00006000
SRCH:    L = INDEX(TEMP,' ');                                               00006010
         IF L = 0                                                           00006020
           THEN DO;                                                         00006030
                L = LENGTH(TEMP);                                           00006040
                SUM1 = SUM1 + L;                                            00006050
                IF SUM1 > LNTH                                              00006060
                  THEN DO;                                                  00006070
                       NU = NU + 1;                                         00006080
                       BROKE(NU) = SUBSTR(BREAKE,1,SUM1-L-1);               00006090
                       NU = NU + 1;                                         00006100
                       BROKE(NU) = SUBSTR(BREAKE,SUM1-L+1);                 00006110
                       RETURN;                                              00006120
                       END;                                                 00006130
                  ELSE DO;                                                  00006140
                       NU = NU + 1;                                         00006150
                       BROKE(NU) = SUBSTR(BREAKE,1,SUM1);                   00006160
                       RETURN;                                              00006170
                       END;                                                 00006180
                END;                                                        00006190
SUM:     SUM1 = SUM1 + (L - 1);                                             00006200
```

```
          IF SUM1 <=LNTH THEN DO;                                          00006210
                              TEMP = SUBSTR(TEMP,L+1);                     00006220
                              SUM1 = SUM1 + 1;                             00006230
                              GO TO SRCH;                                  00006240
                              END;                                         00006250
          NU = NU + 1;                                                     00006260
          BROKE(NU) = SUBSTR(BREAKE,1,SUM1-L);                             00006270
          BREAKE = SUBSTR(BREAKE,SUM1+2-L);                                00006280
          SUM1 = 0;                                                        00006290
          GO TO SUM;                                                       00006300
          END BREAK;                                                       00006310
                                                                           00006320
                                                                           00006330
   /*   THIS PROCEDURE IS CALLED TO DELETE THE LEADING ZEROES FROM A VT */00006340
   /*   NUMBER.  THUS, '004506' BECOMES '4506'.                         */00006350
                                                                           00006360
                                                                           00006370
   DELET:                                                                  00006380
          PROCEDURE;                                                       00006390
          DECLARE                                                          00006400
            M                 FIXED BINARY (31,0),                       00006410
            TEMP              CHARACTER (6) VARYING,                     00006420
            BLANK             CHARACTER (1) INITIAL (' '),               00006430
          ;                                                                00006440
          TEMP = VT;                                                       00006450
   LP:    DO M = 1 TO 6;                                                   00006460
          IF SUBSTR(TEMP,M,1)='0' THEN DO;                                 00006470
                                  IF M=1 THEN VT=BLANK||SUBSTR(VT,2);      00006480
                                         ELSE VT=SUBSTR(VT,1,M-1) ||       00006490
                                                 BLANK              ||     00006500
                                                 SUBSTR(VT,M+1);           00006510
                                  END;                                     00006520
                                ELSE GO TO FIX;                            00006530
          END LP;                                                          00006540
   FIX:   VT = SUBSTR(VT,6-WD+1,WD);                                       00006550
          RETURN;                                                          00006560
          END DELET;                                                       00006570
                                                                           00006580
          END BUSINESS;                                                    00006590
                                                                           00006600
          END ANNSUB;                                                      00006610
```

AUTHOR ANNUAL

```
/*                                                                    */
 *                                                                    *
/*            A N N U A L    A U T H O R    I N D E X                */
/*                                                                    */
/*                                                                    */
/*  PURPOSE: GENERATION OF AUTHOR INDEX FOR AIM/ARM ANNUAL INDEX      */
/*           USES AUTHOR TAPES GENERATED DURING THE YEAR FOR INPUT    */
/*           ONLY THE VT NUMBER, YEAR OF PUBLICATION, AND AUTHOR      */
/*           NAME ARE NEEDED FOR THE INDEX.                           */
/*           FOR NAMES OF INPUT PARAMETERS SEE COMMENT BELOW.         */
/*    SYSIN DATA: THE PARAMETERS TO CONTROL FORMATTING ARE READ IN    */
/*           USING A DATA FORM THUS TO READ IN THE VALUE OF MARGIN    */
/*           A CARD WITH -  MARGIN=    - IS REQUIRED.                 */
/*                                                                    */
/**********************************************************************/
YRAUT: PROCEDURE OPTIONS(MAIN);
BEGIN;
/*
  SORT THE QUARTERLY INDECIES INTO ONE DATA SET
*/
DCL IHESRTA
      ENTRY(CHAR(48),CHAR(34),FIXED BINARY(31,0),FIXED BINARY(15,0)),
      RETCDE FIXED BINARY(15,0),
      SFIELD CHAR(48),
      RFIELD CHAR(34);
SFIELD=' SORT FIELDS=(5,132,CH,A,137,10,CH,A),SIZE=E300 ';
RFIELD=' RECORD TYPE=V,LENGTH=(564,,,171) ';
CALL IHESRTA(SFIELD,RFIELD,61440,RETCDE);
PUT FILE(SYSPRINT) LIST('RETURN CODE=' ,RETCDE);
END;
DCL (MARGIN,VT_WIDTH,DATE,INDEX_PAGE,#_COLS,PRINT_LINE,PAGE_LINES,RAN,
     FIRST_INITIAL,NEXT_INITIAL) FIXED BINARY(31,0),
     VT_ED_SKIP FIXED BINARY(31,0),
      LAST_PRINT FIXED BINARY(31,0),
     COL_SPACE CHAR(20) VAR,
     TOP CHAR(4) INITIAL('LEFT'
     HEADING CHAR (20) VAR;
/*
  OBTAIN PARAMETER FOR THIS INDEX
     MARGIN = WIDTH OF PRINT
     VT_WIDTH = COLUMNS ALLOCATED TO VT NUMBER
     DATE = LAST TWO DIGITS OF PRESENT YEAR
          PROGRAM ALLOWS FOR DOCUMENTS PUBLISHED BETWEEN 1950 AND PRES.
     INDEX_PAGE = FIRST PAGE NUMBER FOR THIS INDEX
     #_COLS = NUMBER OF COLUMNS DESIRED
     PRINT_LINE  = NUMBER OF COLUMNS ACROSS PAGE
     PAGE LINES = LENGTH OF PAGE IN NUMBER OF LINES
     HEADING = WHICH PUBLICATION IS THIS
     TOP = WHICH SIDE OF PAGE IS FIRST PAGE NUMBER
     FIRST_INITIAL = FIRST LINE OF PRINT FOR FIRST PAGE
     NEXT_INITIAL = FIRST LINE OF PRINT ON OTHER THAN FIRST PAGE
     COL_SPACE = BLANKS TO BE PLACE BETWEEN COLUMNS

*/
GET FILE(SYSIN) DATA(MARGIN,VT_WIDTH,DATE,INDEX_PAGE,#_COLS,PRINT_LINE,
     PAGE_LINES,HEADING,TOP,FIRST_INITIAL,NEXT_INITIAL,COL_SPACE) COPY;
       RAN=DATE-50;
 LAST_PRINT=0;
BEGIN;
DCL (INITIAL_LINE,COL_COUNT,IT,PG_L_OR_R,TOTAL_PUB(50:DATE+1),
     YEAR,VT#,J,I,K,LAST,#_VTS,PUBLICATIONS(RAN),K1,LN,LNA,
```

```
      AUTHOR_PUB(50:DATE+1,50),KEY,ITERS,FIRST,QUARTER,MAX_LINES) FIXED 00000640
      BINARY (31,0),                                                   00000650
     (VT1,VT2) FIXED BINARY(31,0),                                     00000660
      AIM_ARM FIXED BINARY(31,0),                                      00000670
      AUTHOR_COUNT FIXED BINARY(31,0),                                 00000680
      SAVE_LINE CHAR(MARGIN),                                          00000690
      AUTHOR_SAVE(10) CHAR(MARGIN),                                    00000700
      OUT_PAGE(PAGE_LINES) CHAR(PRINT_LINE) VAR,                       00000710
       YR_CHR CHAR(2),                                                 00000720
     NAME CHAR(132),                                                   00000730
     REC CHAR(949) VAR,                                                00000740
     PREVIOUS CHAR(132) INITIAL((132)' '),                             00000750
      (LINE,OUT_COL(PAGE_LINES+10),SUBNAME) CHAR(MARGIN),              00000760
      BLANK_LINE CHAR(MARGIN);                                         00000770
DCL ISAVE FIXED BINARY(31,0);                                          00000780
     BLANK_LINE=' ';                                                   00000790
     COL_COUNT=0;                                                      00000800
OPEN FILE(AUTHORS) RECORD INPUT TITLE('SORTOUT');                      00000810
OPEN FILE(OUT) PRINT PAGESIZE(PAGE_LINES) LINESIZE(PRINT_LINE);        00000820
         TOTAL_PUB=0;                                                  00000830
     LNA=1;                                                            00000840
     LN=1;                                                             00000850
COL_COUNT=0;                                                           00000860
INITIAL_LINE=FIRST_INITIAL;                                            00000870
PUT FILE(OUT) PAGE LINE(INITIAL_LINE);                                 00000880
/*                                                                     00000890
   SECTION TO PRINT TOP OF PAGE                                        00000900
*/                                                                     00000910
ON ENDPAGE(OUT) BEGIN;                                                 00000920
        INITIAL_LINE=NEXT_INITIAL;                                     00000930
        IF TOP='LEFT' THEN DO;                                         00000940
              TOP='RITE';                                              00000950
               AIM_ARM=PRINT_LINE-LENGTH(HEADING)-1;                   00000960
              PG_L_OR_R=9;                                             00000970
                        END;                                           00000980
                        ELSE DO;                                       00000990
              TOP='LEFT';                                              00001000
       AIM_ARM=9;                                                      00001010
              PG_L_OR_R=PRINT_LINE-5;                                  00001020
                        END;                                           00001030
        PUT FILE(OUT) PAGE LINE(2);                                    00001040
PUT FILE(OUT) EDIT(INDEX_PAGE,HEADING,'PERSONAL AND INSTITUTIONAL AUTH00001050
RS INDEX')(COLUMN(PG_L_OR_R),F(3),SKIP(0),COLUMN(AIM_ARM),A,SKIP(2),   00001060
X(PRINT_LINE/2-21),A(40));                                             00001070
          PUT FILE(OUT) SKIP(3);                                       00001080
     COL_COUNT=0;                                                      00001090
       OUT_PAGE='    ';                                              00001100
       INDEX_PAGE=INDEX_PAGE+1;                                        00001110
                        END;                                           00001120
                                                                       00001130
/*                                                                     00001140
     WHEN END OF INPUT IS REACHED FINISH LAST PAGE AND PRINT           00001150
*/                                                                     00001160
ON ENDFILE(AUTHORS) BEGIN;                                             00001170
     KEY=3;                                                            00001180
     GO TO FINAL;                                                      00001190
                        END;                                           00001200
#_VTS=(MARGIN-5)/(VT_WIDTH+1);                                         00001210
              KEY=1;                                                   00001220
       OUT_PAGE='    ';                                              00001230
     ON CONVERSION BEGIN;                                              00001240
          PUT FILE(SYSPRINT) DATA(REC);                                00001250
          GO TO INPUT;
```

```
                        END;                                                    00001260
INPUT: READ FILE(AUTHORS) INTO(REC);                                            00001270
PUT FILE(SYSPRINT) EDIT(REC)(SKIP,A);                                           00001280
     YR_CHR=SUBSTR(REC,133,2);                                                  00001290
/*                                                                              00001300
    CHECK YEAR POSITION FOR WRONG CHARACTERS                                    00001310
       IF OBVIOUS MISTAKE CORRECT IT                                            00001320
*/                                                                              00001330
    IF YR_CHR¬='  '&SUBSTR(YR_CHR,1,1)=' ' THEN SIGNAL CONVERSION;              00001340
    IF SUBSTR(YR_CHR,2,1)='L' THEN SUBSTR(REC,158,1)='1';                       00001350
     IF YR_CHR='ND'|YR_CHR='  ' THEN DO;                                        00001360
                        YEAR=DATE+1;                                            00001370
                        GET STRING(REC) EDIT(NAME,VT1,VT2)(A(132),X(2),2 F(3))00001380
                                ;                                               00001390
                                END;                                            00001400
 ELSE GET STRING(REC) EDIT(NAME,YEAR,VT1,VT2)(A(132),F(2),2 F(3));              00001410
IF YEAR<50 THEN DO;                                                             00001420
                    PUT FILE(SYSPRINT) DATA(REC);                               00001430
                    GO TO INPUT;                                                00001440
                      END;                                                      00001450
       VT#=VT1*1000+VT2;                                                        00001460
       IF PREVIOUS=NAME THEN GO TO FILL;                                        00001470
       IF LNA=1 THEN GO TO NOBLANK;                                             00001480
FINAL: OUT_COL(LNA)=BLANK_LINE;                                                 00001490
         LNA=LNA+1;                                                             00001500
NOBLANK: MAX_LINES=PAGE_LINES-INITIAL_LINE;                                     00001510
          AUTHOR_COUNT=0;                                                       00001520
/*                                                                              00001530
   BREAK UP AUTHOR NAME TO FIT IN PRINT COLUMN                                  00001540
*/                                                                              00001550
   DO J=MARGIN,J+K,J+K,J+K,J+K,J+K,J+K;                                         00001560
       SUBNAME=SUBSTR(PREVIOUS,J-MARGIN+1,MARGIN);                              00001570
       I=0;                                                                     00001580
                IF SUBSTR(SUBNAME,MARGIN,1)¬=' ' THEN DO;                       00001590
                DO I=1 TO MARGIN-1;                                             00001600
                 IF SUBSTR(SUBNAME,MARGIN-I,1)=' '|SUBSTR(SUBNAME,MARGIN-       00001610
                        I,1)='-' THEN GO TO BREAK;                              00001620
                    END;                                                        00001630
  PUT FILE(SYSPRINT) LIST(' AUTHOR NAME',NAME,'  VT# = ',VT#);                  00001640
                                          END;                                  00001650
BREAK: SUBNAME=SUBSTR(SUBNAME,1,MARGIN-I);                                      00001660
       K=INDEX(SUBNAME,'  ');                                                   00001670
        IF K=1 THEN GO TO DOCUM;                                                00001680
       IF K=0 THEN K=MARGIN-I;                                                  00001690
    PUT STRING(LINE) EDIT(SUBNAME)(A(K));                                       00001700
/*                                                                              00001710
   SAVE AUTHOR NAME TO BE USED IF NESSESARY IN CONTINUATION COLUMN              00001720
*/                                                                              00001730
    AUTHOR_COUNT=AUTHOR_COUNT+1;                                                00001740
    AUTHOR_SAVE(AUTHOR_COUNT)=LINE;                                             00001750
     OUT_COL(LNA)=LINE;                                                         00001760
    LNA=LNA+1;                                                                  00001770
   END;                                                                         00001780
 DOCUM: J=0;                                                                    00001790
                IF KEY=1 THEN DO;                                               00001800
                    KEY=0;                                                      00001810
                    PREVIOUS=NAME;                                              00001820
                    GO TO FILL;                                                 00001830
                END;                                                            00001840
IF KEY¬=3 THEN KEY=0;                                                           00001850
       IF LNA>MAX_LINES THEN DO;                                                00001860
                            LNA=LNA-AUTHOR_COUNT;                               00001870
```

```
                                CALL PRINT;                                    00001880
                                DO LNA=1 TO AUTHOR_COUNT;                      00001890
                                   OUT_COL(LNA)=AUTHOR_SAVE(LNA);              00001900
                                END;                                           00001910
                             END;                                              00001920
                                                                               00001930
          DO I=50 TO DATE;                                                     00001940
           IF TOTAL_PUB(I)=0 THEN GO TO NONE;                                  00001950
          J=J+1;                                                               00001960
           PUBLICATIONS(J)=I;                                                  00001970
NONE:     END;                                                                 00001980
      IF TOTAL_PUB(DATE+1)=0 THEN DO;                                          00001990
                                K=1;                                           00002000
                                K1=PUBLICATIONS(1);                            00002010
                                END;                                           00002020
                             ELSE DO;                                          00002030
                                K=DATE+1;                                      00002040
                                K1=K;                                          00002050
                                END;                                           00002060
OUTPUT: FIRST=1;                                                               00002070
          LAST=#_VTS;                                                          00002080
           THOUSANDS=FIXED(AUTHOR_PUB(K1,1)/1000);                             00002090
      IF K=DATE+1 THEN GO TO JUMP;                                             00002100
          ITERS=#_VTS;                                                         00002110
 /*                                                                            00002120
      FILL IN PRINT LINE FOR A PARTICULAR YEAR                                 00002130
 */                                                                            00002140
       PUT STRING(LINE) EDIT(K1,':',AUTHOR_PUB(K1,1),(',',AUTHOR_PUB(K1,      00002150
       I) DO I=2 TO ITERS WHILE((FIXED(AUTHOR_PUB(K1,I)/1000)=THOUSANDS)       00002160
      &(I<=TOTAL_PUB(K1))))(F(2),A(1),X(2),(ITERS)(F(VT_WIDTH),A(1)));         00002170
 IF I-1=#_VTS&AUTHOR_PUB(K1,I)/1000=THOUSANDS THEN SUBSTR(LINE,ITERS*          00002180
 (VT_WIDTH+1)+5,1)=',';                                                        00002190
 LOOP1: OUT_COL(LNA)=LINE;                                                     00002200
          IF KEY=5 THEN DO;                                                    00002210
             SAVE_LINE=LINE;                                                   00002220
                    ISAVE=I;                                                   00002230
                      KEY=0;                                                   00002240
                   DO I=1 TO AUTHOR_COUNT-1;                                   00002250
                     OUT_COL(I)=AUTHOR_SAVE(I);                                00002260
                   END;                                                        00002270
                   I=INDEX(AUTHOR_SAVE(AUTHOR_COUNT),'          ');           00002280
                     IF I=0 THEN DO;                                           00002290
                      OUT_COL(AUTHOR_COUNT)=AUTHOR_SAVE(AUTHOR_COUNT);         00002300
                     I=AUTHOR_COUNT+1;                                         00002310
                      OUT_COL(I)='(CONTINUED)';                                00002320
                     END;                                                      00002330
                          ELSE DO;                                             00002340
                     OUT_COL(AUTHOR_COUNT)=SUBSTR(AUTHOR_SAVE(                 00002350
                       AUTHOR_COUNT),1,1)||'(CONTINUED)';                      00002360
                     I=AUTHOR_COUNT;                                           00002370
                   END;                                                        00002380
                   QUARTER=I+1;                                                00002390
          IF SUBSTR(SAVE_LINE,1,2)='  ' THEN PUT STRING(SAVE_LINE)             00002400
                   EDIT(K1,':',SUBSTR(SAVE_LINE,4))(F(2),A(1),A);              00002410
                     OUT_COL(QUARTER)=SAVE_LINE;                               00002420
                   LNA=I+1;                                                    00002430
                   I=ISAVE;                                                    00002440
                                END;                                           00002450
      LN=LNA;                                                                  00002460
       LNA=LNA+1;                                                              00002470
 /*                                                                            00002480
      IF THIS IS CONTINUATION OF AN AUTHOR IN A SECOND COLUMN THEN KEY =5      00002490
        AND THE AUTHOR NAME NEEDS TO BE PLACED AT TOP OF COLUMN
```

```
*/                                                                  00002500
    IF LNA>=MAX_LINES THEN DO;                                      00002510
                          KEY=5;                                    00002520
                 QUARTER=I;                                         00002530
                          CALL PRINT;                               00002540
                        OUT_COL(1)=BLANK_LINE;                      00002550
                 I=QUARTER;                                         00002560
                          END;                                      00002570
IF I>TOTAL_PUB(K1) THEN GO TO ENDK;                                 00002580
      LAST=I-1;                                                     00002590
         THOUSANDS=FIXED(AUTHOR_PUB(K1,I)/1000);                    00002600
    FIRST=LAST+1;                                                   00002610
      IF FIRST>TOTAL_PUB(K1) THEN GO TO ENDK;                       00002620
      LAST=LAST+#_VTS;                                              00002630
JUMP: IF LAST>TOTAL_PUB(K1) THEN LAST=TOTAL_PUB(K1);                00002640
    ITERS=LAST-FIRST+1;                                             00002650
      IF J=I THEN PUT STRING(LINE) EDIT(K1,':')(F(2),A(1));         00002660
                                                                    00002670
/*                                                                  00002680
    FILL IN PRINT LINE FOR A PARTICULAR YEAR & DELETE YEAR IF CONTIUAT00002690
*/                                                                  00002700
      PUT STRING(LINE) EDIT(AUTHOR_PUB(K1,FIRST),(',',AUTHOR_PUB(K1,I)00002710
    DO I=FIRST+1 TO LAST WHILE(FIXED(AUTHOR_PUB(K1,I)/1000)=THOUSANDS00002720
    ))(X(5),(ITERS)(F(VT_WIDTH),A(1)));                             00002730
IF I-1=LAST&AUTHOR_PUB(K1,I)/1000=THOUSANDS&LAST¬=TOTAL_PUB(K1) THEN 00002740
SUBSTR(LINE,ITERS*(VT_WIDTH+1)+5,1)=',';                            00002750
      IF K=DATE+1&FIRST=1 THEN SUBSTR(LINE,1,3)='NO:';              00002760
      GO TO LOOP1;                                                  00002770
ENDK: IF K=DATE+1 THEN K=1;                                         00002780
                  ELSE K=K+1;                                       00002790
    IF K<=J THEN DO;                                                00002800
                  K1=PUBLICATIONS(K);                               00002810
                  GO TO OUTPUT;                                     00002820
                 END;                                               00002830
       PREVIOUS=NAME;                                               00002840
   IF KEY=3 THEN CALL PRINT;                                        00002850
IF KEY=3 THEN GO TO AUT;                                            00002860
 LN=LNA;                                                            00002870
   TOTAL_PUB=0;                                                     00002880
/*                                                                  00002890
   FILL UP ARRAYS FOR A PARTICULAR AUTHOR BEFORE FORMATTING A LINE  00002900
*/                                                                  00002910
FILL: TOTAL_PUB(YEAR)=TOTAL_PUB(YEAR)+1;                            00002920
      AUTHOR_PUB(YEAR,TOTAL_PUB(YEAR))=VT#;                         00002930
      GO TO INPUT;                                                  00002940
 AUT: PUT FILE(SYSPRINT) DATA(INDEX_PAGE);                          00002950
/*                                                                  00002960
   IF COLUMN HAS BEEN FILLED THIS PROCEDURE WILL BE CALLED THE COLUMN 00002970
      NEEDS TO BE FIT ON PAGE AND LEFT OVER SHIFTED TO TOP OF NEXT C00002980
*/                                                                  00002990
PRINT: PROCEDURE;                                                   00003000
      COL_COUNT=COL_COUNT+1;                                        00003010
   LAST_PRINT=LN;                                                   00003020
  IF COL_COUNT>1 THEN DO I=1 TO LAST_PRINT;                         00003030
            OUT_PAGE(I)=OUT_PAGE(I)||COL_SPACE||OUT_COL(I);         00003040
                       END;                                         00003050
                  ELSE                                              00003060
                       DO I=1 TO LAST_PRINT;                        00003070
             OUT_PAGE(I)=OUT_PAGE(I)||'      '||OUT_COL(I);         00003080
            END;                                                    00003090
  IF LAST_PRINT¬=MAX_LINES THEN DO;                                 00003100
  IF COL_COUNT>1 THEN DO I=LAST_PRINT+1 TO MAX_LINES;               00003110
            OUT_PAGE(I)=OUT_PAGE(I)||COL_SPACE||BLANK_LINE;
```

```
                        END;                                                    00003120
                      ELSE                                                      00003130
                       DO I=LAST_PRINT+1 TO MAX_LINES;                          00003140
                 OUT_PAGE(I)=OUT_PAGE(I)||'      '||BLANK_LINE;                00003150
               END;                                                             00003160
    END;                                                                        00003170
      IF COL_COUNT=#_COLS THEN DO I=1 TO MAX_LINES;                             00003180
                               PUT FILE(OUT) EDIT(OUT_PAGE(I))(SKIP,A);         00003190
                                 END;                                           00003200
      IF KEY=3&COL_COUNT¬=#_COLS THEN DO I=1 TO MAX_LINES;                      00003210
                               PUT FILE(OUT) EDIT(OUT_PAGE(I))(SKIP,A);         00003220
                                 END;                                           00003230
       IF COL_COUNT=#_COLS|KEY=3 THEN SIGNAL ENDPAGE(OUT);                      00003240
       IF KEY=3 THEN GO TO AUT;                                                 00003250
       LNA=1;                                                                   00003260
  END PRINT;                                                                    00003270
   END YRAUT;                                                                   00003280
```

SUPPORTING SERVICES ANNUAL

```
/*********************************************************************/
/*                                                                   */
/*      A N N U A L   S U P P O R T I N G   S E R V I C E S          */
/*                  I N D E X   P R O G R A M                        */
/*                                                                   */
/*                                                                   */
/*  PURPOSE:  THIS PROGRAM PRODUCES THE COMBINED AIM/ARM SUPPORTING  */
/*            SERVICES INDEX FOR THE ANNUAL PUBLICATION.  THE INPUT  */
/*            TO THE PROGRAM IS THE 4 QUARTERLY DATA SETS OF AIM     */
/*            (EVEN THOUGH THE QUARTERLY AIM PUBLICATIONS DO NOT     */
/*            HAVE THIS INDEX, WE RUN THE QUARTERLY PROGRAM WITH THE */
/*            AIM DATA TO MAKE A DATA SET WITH THE SHORTENED RECORDS)*/
/*            AND THE 4 QUARTERLY ARM DATA SETS.  WE CONCATENATE     */
/*            ALL 8 TOGETHER AND SORT USING THE 360 SORT/MERGE TO    */
/*            A TEMPORARY DISK FILE.  THE DISK FILE IS PRINTED IN    */
/*            A FORMAT DICTATED BY THE VOCATIONAL CENTER ON AN OUTPUT*/
/*            FILE AND WE HAVE AN INDEX.  THE DISK FILE IS DELETED,  */
/*            BUT ALL 8 TAPE DATA SETS ARE KEPT.                     */
/*                                                                   */
/*  DEFINITION OF VARIABLES:                                         */
/*     1. AFIELD -- 3-DIGIT CODE READ FROM THE DISK RECORDS.         */
/*     2. BLK -- NUMBER OF PIECES INTO WHICH A CODE NAME MUST BE     */
/*               BROKEN UP SO THAT IT WILL FIT IN A COLUMN OF OUTPUT.*/
/*     3. CNT -- THIS IS AN ARRAY OF NUMBERS, EACH DENOTING THE      */
/*               NUMBER OF DOCUMENTS UNDER A PARTICULAR PUBLICATION  */
/*               DATE. IF P IS THE ARRAY OF DATES FOR A SUB-CODE,    */
/*               THEN UNDER DATE P(J) THERE ARE CNT(J) DOCUMENTS TO  */
/*               BE LISTED IN THE INDEX.  EACH CNT MUST BE LESS THAN */
/*               OR EQUAL TO 200 FOR THIS PROGRAM'S CURRENT SETUP.   */
/*               IF IT EXCEDES 200, CHANGE THE DIMENSIONS ON         */
/*               V RIABLES FV AND V.                                 */
/*     4. P -- ARRAY OF PUBLICATION DATES OR YEAR NUMBERS FOR A      */
/*             SUB-CODE, WHEN DOCUMENTS WERE WRITTEN ABOUT THE SUB-  */
/*             CODE NAME.                                            */
/*               E NUMBER OF DATES RELATED TO A SUB-CODE.  THUS, FOR */
/*               OME SUB-CODE, THERE IS INFORMATION ABOUT IT IN      */
/*               CUMENTS IN THE YEARS P(1), P(2),..., P(I). -I- MUST */
/*             BE LESS THAN OR EQUAL   15 FOR THE PROGRAM'S CURRENT  */
/*             SET-UP.  IF IT EXCEDES 15, CHANGE THE DIMENSIONS ON   */
/*             VARIABLES P, CNT, FV, V.                              */
/*     6. FV -- THESE ARE THE DOCUMENT NUMBERS CONTAING INFORMATION  */
/*               ABOUT A SUB-CODE NAME.  -FV- IS A TWO DIMENSIONAL   */
/*               ARRAY; THE FIRST SUBSCRIPT -J- REFERENCES THE       */
/*               PUBLICATION YEAR -P(J)- OF THE DOCUMENT, THE SECOND */
/*               SUBSCRIPT -L- REFERENCES A PARTICULAR DOCUMENT NO.  */
/*               UNDER THIS YEAR.                                    */
/*     7. V -- SAME AS -FV- ONLY FV IS NUMERIC AND V IS CHARACTER.   */
/*     8. COL -- NUNBER OF COLUMNS IN THE INDEX PRINTOUT.            */
/*     9. DUMMY -- STRING INTO WHICH A RECORD IS READ.               */
/*    10. FLD -- ARRAY OF SUB-CODE NAMES.                            */
/*    11. FRST -- PAGE NUMBER OF TITLE PAGE.                         */
/*    12. FVT -- A SINGLE NUMERIC DOCUMENT NUMBER.                   */
/*    13. HEAD -- HEADING WHICH TOPS EVERY PAGE AND IS FOUND ON THE  */
/*                SAME LINE AS THE INDEX PAGE NUMBER.  ITS FORM IS   */
/*                'AIM/ARM 69-70'.                                   */
/*    14. IHESRTA -- CALL TO SORT PROGRAM.                           */
/*    15. IMAX -- INDICATES ALL COLUMNS OF A PAGE ARE FULL.          */
/*    16. IMP -- INDICATES IF WE HAVE PRINTED THE LAST BLOCK OF DATA */
/*               AND THEREFORE, IF THE INDEX IS THROUGH.  IMP=1 SAYS */
/*               WE ARE THROUGH; IMP=0 SAYS MORE DATA TO COME.       */
/*    17. IND -- INDICATES WHAT COLUMN ON A PAGE IS BEING FILLED.    */
/*    18. KNT -- IS THE NUMBER OF LINES OF DATA IN A BLOCK.  BY BLOCK*/
```

```
/*                I MEAN A PUBLICATION DATE AND ALL ITS DOCUMENT          */00000640
/*                NUMBERS LISTED BY THOUSANDS.  TO LIST BY THOUSANDS,     */00000650
/*                WE LIST ALL VT NUMBERS 0 < VT < 999. THEN STARTING      */00000660
/*                ON A NEW LINE WE LIST ALL VT'S WITH 1000<VT#<1999.      */00000670
/*                THIS CONTINUES UNTIL ALL DOCUMENT NUMBERS ARE           */00000680
/*                                                                        */00000690
/*                LISTED.                                                 */00000700
/*     19. LIM -- MAXIMUM NUMBER OF DOCUMENT NUMBERS THAT WILL FIT        */00000710
/*                IN A COLUMN OF SOME WIDTH.                              */00000720
/*     20. LINEH1 -- LINE ON THE TITLE PAGE ON WHICH THE PAGE NUMBER      */00000730
/*                   AND -HEAD- APPEAR.                                   */00000740
/*     21. LINEH2 -- SAME AS -LINEH1- BUT FOR ALL SUCCEEDING PAGES.       */00000750
/*     22. LINES1 -- LINE ON THE TITLE PAGE WHICH IS RIGHT ABOVE THE      */00000760
/*                   INDEX PRINTOUT.                                      */00000770
/*     23. LINES2 -- SAME AS -LINES1- BUT FOR ALL SUCCEEDING PAGES.       */00000780
/*     24. LINET -- LINE ON ALL PAGES WHERE THE TITLE IS PRINTED.         */00000790
/*     25. LN -- IS THE CURRENT LINE BEING FILLED ON A PAGE. IT IS A      */00000800
/*                CHECK TO MAKE SURE THE PAGE LENGTH IS NOT EXCEDED.      */00000810
/*     26. LNE -- COUNTS THE NUMBER OF VT NUMBERS ON A LINE SO THE        */00000820
/*                COUNT DOES NOT EXCEDE -LIM-, THE LIMIT.                 */00000830
/*     27. LTH -- LENGTH OF THE INDEX TITLE.                              */00000840
/*     28. MARG -- LEFT-HAND MARGIN OF AN INDEX PAGE.                     */00000850
/*     29. MG -- MARGINS OF THE COLUMN, EACH MEASURED FROM COLUMN 1.      */00000860
/*     30. MT -- DESIGNATES THE INTERVAL OF THE LAST VT NUMBER TO BE      */00000870
/*                PUT IN THE INDEX UNDER A PARTICULAR YEAR FOR SOME       */00000880
/*                CODE NAME.  MT * 1000 IS THE START OF THE INTERVAL,     */00000890
/*                WHERE MT = 0, 1, 2,... AND MT * 1000 + 999 ENDS THE     */00000900
/*                INTERVAL.                                               */00000910
/*     31. NN -- COUNTS THE LINE NUMBERS BEING FILLED IN A COLUMN.        */00000920
/*     32. OUTS -- OUTPUT FILE FOR INDEX PRINTOUT.                        */00000930
/*     33. PAGE -- PAGE NUMBER OF INDEX.                                  */00000940
/*     34. PREV -- CURRENT CODE NAME WHOSE DATA HAS JUST BEEN PUT IN      */00000950
/*                THE INDEX.                                              */00000960
/*     35. PUB -- PUBLICATION YEAR OF A RECORD ON THE SUBSET FILE.        */00000970
/*     36. QUE -- TITLE PAGE OF INDEX WILL BE ON THE LEFT ('L') OR        */00000980
/*                RIGHT ('R') SIDE OF THE PUBLICATION.                    */00000990
/*     37. RETCDE -- RETURN CODE RETURNED BY THE SORT INDICATING          */00001000
/*                   SUCCESS OR FAILURE.                                  */00001010
/*     38. RFIELD -- DESCRIBES TO SORT THE RECORDS OF THE SUBSET FILE.    */00001020
/*     39. SFIELD -- TELLS SORT WHAT PART(S) OF THE RECORDS TO SORT       */0001030
/*                   UNDER.                                               1040
/*     40. SZ  -- RIGHT-HAND MARGIN OF THE PAGE.                        *  001050
/*     41. SKPH -- TELLS WHERE -HEAD- SHOULD START ON ITS OUTPUT LI     */00001060
/*     42. SKPP -- GIVES THE STARTING COLUMN FOR THE PAGE NUMBER ON       */00001070
/*                 ITS OUTPUT LINE.                                       */00001080
/*     43. SORTIN -- ALL 8 DATA SETS OF THE SUBSET FILE TO BE SORTED.     */00001090
/*     44. SORTOUT -- SORTED RECORDS ON DISK.                             */00001100
/*     45. SIZE -- MAXIMUM LENGTH OF OUTPUT PAGE.                         */00001110
/*     46. TITLE -- TITLE OF THE INDEX.                                   */00001120
/*     47. VT -- A DOCUMENT NUMBER FROM A RECORD OF THE SUBSET FILE.      */00001130
/*     48. WD -- NUMBER OF SIGNIFICANT DIGITS IN THE VT NUMBER.           */00001140
/*     49. WIDTH -- COLUMN WIDTH.                                         */00001150
/*     50. COL# -- IS A 2-DIMENSIONAL ARRAY REPRESENTING A PAGE OF        */00001160
/*                 OUTPUT FROM BELOW THE HEADING.  THE FIRST SUBSCRIPT     */00001170
/*                 CALLS A COLUMN AND THE SECOND SUBSCRIPT CALLS A         */00001180
/*                 LINE IN THE COLUMN.                                    */00001190
/*     51. PC -- PIECES INTO WHICH A CODE NAME IS BROKEN UP SO THAT       */00001200
/*               IT WILL FIT IN A COLUMN.                                 */00001210
/*     52. TEMP_OUT -- FOR SOME SUB-CODE, THIS IS THE OUTPUT SET-UP       */00001220
/*                     FOR ALL THE VT NUMBERS UNDER ONE PUBLICATION       */00001230
/*                     DATE. -TEMP_OUT- IS PUT INTO -COL#- ONCE THE       */00001240
/*                     BLOCK IS COMPLETE.                                 */00001250
/*                                                                        */
```

```
/*  INPUT TO THE PROGRAM:                                                  */00001260
/*    -SORTOUT-, ALL RECORDS FROM THE SUBSET FILE IN SORTED ORDER          */00001270
/*      WE LIST THIS FILE IN A FORMAT AND AN ANNUAL INDEX IS PRODUCED      */00001280
/*      FOR US.                                                            */00001290
/*    -SYSIN-, INPUT ON CARDS                                              */00001300
/*      THIS IS A CONTINUOUS STREAM OF DATA.  THE VARIABLES BELOW ARE      */00001310
/*      GIVEN VALUES ON THE CARDS IN THE SAME ORDER AS THEY ARE            */00001320
/*      LISTED BELOW.                                                      */00001330
/*            -PAGE-          F(4)                                         */00001340
/*            -SIZE-          F(4)                                         */00001350
/*            -WIDTH-         F(4)                                         */00001360
/*            -WD-            F(4)                                         */00001370
/*            -MARG-          F(4)                                         */00001380
/*            -SZ-            F(4)                                         */00001390
/*            -LINEH1-        F(4)                                         */00001400
/*            -LINES1-        F(4)                                         */00001410
/*            -LINEH2-        F(4)                                         */00001420
/*            -LINET-         F(4)                                         */00001430
/*            -LINES2-        F(4)                                         */00001440
/*            -COL-           F(4)                                         */00001450
/*            -MG- ARRAY      COL * F(4)                                   */00001460
/*            -LTH-           F(4)                                         */00001470
/*            -TITLE-         A(LTH)                                       */00001480
/*            -HEAD-          A(13)                                        */00001490
/*            -QUE-           A(1)                                         */00001500
/*      THESE VARIABLES WERE DEFINED IN THE PREVIOUS SECTION AND           */00001510
/*      THEIR VALUES CAN BE OBTAINED FROM THE TYPED WRITE-UP OR THE        */00001520
/*      VOCATIONAL CENTER.  THE VARIABLES DEFINE AN OUTPUT PAGE.           */00001530
/*                                                                         */00001540
/*  SPECIAL NOTE:                                                          */00001550
/*      PRINT THE INDEX ON THE PRINTER UNTIL IT HAS BEEN CHECKED BY        */00001560
/*      THE VOCATIONAL CENTER.  THEN PRINT IT ON MAGNETIC TAPE AND         */00001570
/*      LIST THE TAPE ON THE 360/50 USING THE TN CHAIN FOR UPPER AND       */00001580
/*      LOWER CASE LETTERS.  THIS IS THE GOOD COPY OF THE INDEX.           */00001590
/*                                                                         */00001600
/*************************************************************************/00001610
                                                                           00001620
ANNVCT:  PROCEDURE OPTIONS (MAIN);                                         00001630
         DECLARE                                                           00001640
           AFIELD              FIXED BINARY (31,0),                        00001650
           BLK                 FIXED BINARY (31,0),                        00001660
           CNT(15)             FIXED BINARY                                00001670
           COL                 FIXED BINAR      ,0),                       00001680
           DUMMY               CHARACTER (18),                             00001690
           FLD(0:13)           CHARACTER (132) VARYING INITIAL             00001700
                               ('AGRICULTURAL EDUCATION',                  00001710
                                'BUSINESS AND OFFICE EDUCATION',           00001720
                                'DISTRIBUTIVE EDUCATION',                  00001730
                                'HEALTH OCCUPATIONS EDUCATION',            00001740
                                'HOME ECONOMICS EDUCATION',                00001750
                                'INDUSTRIAL ARTS EDUCATION',               00001760
                                'MANPOWER ECONOMICS',                      00001770
                                'OCCUPATIONAL PSYCHOLOGY',                 00001780
                                'OCCUPATIONAL SOCIOLOGY',                  00001790
                                ' ',                                       00001800
                                'TRADE AND INDUSTRIAL EDUCATION',          00001810
                                ' ',                                       00001820
                               'GENERAL VOCATIONAL AND TECHNICAL EDUCATION',00001830
                                'TECHNICAL EDUCATION'),                    00001840
           FRST                FIXED BINARY (31,0),                        00001850
           FV(15,200)          FIXED BINARY (31,0) INITIAL ((3000)0),      00001860
           FVT                 FIXED BINARY (31,0),                        00001870
```

```
        HEAD            CHARACTER (13),                                 00001880
        I               FIXED BINARY (31,0),                            00001890
        IHESRTA         ENTRY(CHARACTER(53),CHARACTER(27),              00001900
                             FIXED BINARY(31,0),FIXED BINARY            00001910
                             (15,0)),                                   00001920
        IMAX            FIXED BINARY (31,0),                            00001930
        IMP             FIXED BINARY (31,0),                            00001940
        IND             FIXED BINARY (31,0),                            00001950
        J               FIXED BINARY (31,0),                            00001960
        K               FIXED BINARY (31,0),                            00001970
        KNT             FIXED BINARY (31,0),                            00001980
        L               FIXED BINARY (31,0),                            00001990
        LIM             FIXED BINARY (31,0),                            00002000
        LINEH1          FIXED BINARY (31,0),                            00002010
        LINEH2          FIXED BINARY (31,0),                            00002020
        LINES1          FIXED BINARY (31,0),                            00002030
        LINES2          FIXED BINARY (31,0),                            00002040
        LINET           FIXED BINARY (31,0),                            00002050
        LN              FIXED BINARY (31,0),                            00002060
        LNE             FIXED BINARY (31,0),                            00002070
        LTH             FIXED BINARY (31,0),                            00002080
        MARG            FIXED BINARY (31,0),                            00002090
        MG(3)           FIXED BINARY (31,0),                            00002100
        MT              FIXED BINARY (31,0),                            00002110
        N               FIXED BINARY (31,0),                            00002120
        NN              FIXED BINARY (31,0),                            00002130
        OUTS            FILE STREAM,                                    00002140
        P(15)           CHARACTER (2),                                  00002150
        PAGE            FIXED BINARY (31,0),                            00002160
        PREV            CHARACTER (132) VARYING,                        00002170
        PUB             CHARACTER (2),                                  00002180
        QUE             CHARACTER (1),                                  00002190
        RETCDE          FIXED BINARY (15,0),                            00002200
        RFIELD          CHARACTER (27),                                 00002210
        SFIELD          CHARACTER (53),                                 00002220
        SIZE            FIXED BINARY (31,0),                            00002230
        SKPH            FIXED BINARY (31,0),                            00002240
        SKPP            FIXED BINARY (31,0),                            00002250
        SORTIN          FILE SEQUENTIAL RECORD,                         00002260
        SORTOUT         FILE SEQUENTIAL RECORD,                         00002270
        SZ              FIXED BINARY (31,0),                            00002280
        TITLE           CHARACTER (50) VARYING,                         00002290
        V(15,200)       CHARACTER (6) VARYING,                          00002300
        VT              CHARACTER (6) VARYING,                          00002310
        WD              FIXED BINARY (31,0),                            00002320
        WIDTH           FIXED BINARY (31,0),                            00002330
    ;                                                                   00002340
                                                                        00002350
                                                                        00002360
/*  SORT THE 8 DATA SETS OF THE SUBSET FILE BY ASCENDIND ORDER SUB-COD00002370
    THEN BY ASCENDING ORDER PUBLICATION YEAR, AND FINALLY BY ASCENDINGO0002380
    ORDER VT NUMBER.  PRINT OUT THE RETURN CODE AND THE SORTED RECORDS00002390
    FOR A CHECK.                                                        */00002400
                                                                        00002410
                                                                        00002420
    SFIELD = ' SORT FIELDS=(1,3,CH,A,4,2,CH,A,6,6,CH,A),SIZE=E9999      00002430
';                                                                      00002440
    RFIELD = ' RECORD TYPE=F,LENGTH=(18) ';                             00002450
    CALL IHESRTA (SFIELD,RFIELD,45056,RETCDE);                          00002460
    PUT FILE (SYSPRINT) EDIT (RETCDE)  (SKIP(3),F(3));                  00002470
                                                                        00002480
                                                                        00002490
```

```
/*   READ IN THE PAGE SPECIFICATIONS OF AN INDEX PAGE.                    */00002500
                                                                            00002510
                                                                            00002520
      GET FILE (SYSIN) EDIT (PAGE,SIZE,WIDTH,WD,MARG,SZ,LINEH1,LINES1       00002530
        ,LINEH2,LINET,LINES2,COL,(MG(I) DO I = 1 TO COL),LTH,TITLE,         00002540
        HEAD,QUE)                                                           00002550
        (13 F(4),(COL) F(4),A(LTH),A(13),A(1));                             00002560
                                                                            00002570
                                                                            00002580
/*   BEGIN PRINTING OF THE INDEX.                                         */00002590
                                                                            00002600
                                                                            00002610
BUSINESS:                                                                   00002620
      BEGIN;                                                                00002630
      DECLARE                                                               00002640
        COL#(COL,SIZE-LINES2)  CHARACTER (WIDTH),                           00002650
        HEADR                  ENTRY (FIXED BINARY (31,0)),                 00002660
        PC(4)                  CHARACTER (WIDTH) VARYING,                   00002670
        TEMP_OUT(SIZE-LINES2)  CHARACTER (WIDTH) VARYING,                   00002680
      ;                                                                     00002690
                                                                            00002700
                                                                            00002710
/*   WHEN ALL OF THE SORTED RECORDS HAVE BEEN READ, WE MUST PRINT OUT      00002720
     THAT LAST PAGE BEING PUT TOGETHER.                                   */00002730
                                                                            00002740
                                                                            00002750
      ON ENDFILE (SORTOUT) BEGIN;                                           00002760
        IMP = 1;                                                            00002770
        FLD(K) = 'ZZ';                                                      00002780
        GO TO TEST;                                                         00002790
        END;                                                                00002800
                                                                            00002810
                                                                            00002820
/*   MAKE INITIALIZATIONS FOR PRINTING THE INDEX.                         */00002830
                                                                            00002840
                                                                            00002850
BLKOT:  DO I = 1 TO SIZE-LINES2;                                            00002860
        DO J = 1 TO COL;                                                    00002870
        COL#(J,I) = ' ';                                                    00002880
        END;                                                                00002890
        END BLKOT;                                                          00002900
        OPEN FILE (SORTOUT),                                                00002910
             FILE (OUTS) OUTPUT STREAM PRINT PAGESIZE(SIZE + 4)             00002920
        ;                                                                   00002930
        IND = 1;                                                            00002940
        FRST = PAGE;                                                        00002950
        IMAX = COL + 1;                                                     00002960
        LIM = (WIDTH - 4)/(WD + 2);                                         00002970
        PUT DATA (LIM);                                                     00002980
        NN = 0;                                                             00002990
        LN = 9;                                                             00003000
        IMP = 0;                                                            00003010
        IF QUE = 'L' THEN DO;                                               00003020
                          SKPP = MARG;                                      00003030
                          SKPH = SZ - 12;                                   00003040
                          END;                                              00003050
                     ELSE DO;                                               00003060
                          SKPP = SZ - 3;                                    00003070
                          SKPH = MARG;                                      00003080
                          END;                                              00003090
                                                                            00003100
                                                                            00003110
```

```
   /*  PUT A TITLE ON THE OUTPUT FILE FOR THE INDEX TITLE PAGE.          */00003120
                                                                          00003130
                                                                          00003140
          PUT FILE (OUTS) EDIT (PAGE,HEAD)                                00003150
            (PAGE,LINE(LINEH1),COLUMN(SKPP),F(4),SKIP(0),COLUMN(SKPH),    00003160
            A(13));                                                       00003170
          PUT FILE (OUTS) SKIP (LINES1-LINEH1);                           00003180
                                                                          00003190
                                                                          00003200
   /*  READ IN THE FIRST  CORD OF THE SORTED FILE.                       */00003210
                                                                          00003220
                                                                          00003230
   READ:  READ FILE (SORTOUT) INTO (DUMMY);                               00003240
          GET STRING (DUMMY) EDIT (AFIELD,PUB,VT)                         00003250
              (F(3,0),A(2),A(6));                                         00003260
          GET STRING (DUMMY) EDIT (FVT)                                   00003270
              (X(5),F(6,0));                                              00003280
          CALL DELET;                                                     00003290
          IF AFIELD = 495 THEN K = 13;                                    00003300
                          ELSE K = (AFIELD-400)/10;                       00003310
                                                                          00003320
                                                                          00003330
   /*  WE WILL BE READING -SORTOUT- UNTIL ALL THE INFORMATION FOR A       00003340
       SUB-CODE IS OBTAINED.  WE MAKE CERTAIN INITIALIZATIONS FOR         00003350
       THIS.                                                             */00003360
                                                                          00003370
                                                                          00003380
   INITL: I = 1;                                                          00003390
          J = 1;                                                          00003400
          PREV = FLD(K);                                                  00003410
          IF PUB = '  ' THEN PUB = 'ND';                                  00003420
          CNT(1) = 1;                                                     00003430
          P(1) = PUB;                                                     00003440
          V(1,1) = VT;                                                    00003450
          FV(1,1) = FVT;                                                  00003460
                                                                          00003470
                                                                          00003480
   /*  READ IN A RECORD FROM -SORTOUT-.                                  */00003490
                                                                          00003500
                                                                          00003510
   RD_ON: READ FILE (SORTOUT) INTO (D                                     00003520
          GET STRING (DUMMY) EDIT (AFIELD,PUB,VT)                         00003530
              (F(3,0),A(2),A(6));                                         00003540
          GET STRING (DUMMY) EDIT (FVT)                                   00003550
              (X(5),F(6,0));                                              00003560
          CALL DELET;                                                     00003570
          IF PUB = '  ' THEN PUB = 'ND';                                  00003580
          IF AFIELD = 495 THEN K = 13;                                    00003590
                          ELSE K = (AFIELD-400)/10;                       00003600
                                                                          00003610
                                                                          00003620
   /*  IF THE SAME SUB-CODE IS STILL BEING USED, THEN TAKE THIS PATH.    */00003630
                                                                          00003640
                                                                          00003650
   TEST:  IF FLD(K) = PREV                                                00003660
            THEN DO;                                                      00003670
                                                                          00003680
                                                                          00003690
            /* IF THE DATE IS STILL THE SAME, ADD THIS DOCUMENT           00003700
               TO THE NEXT POSITION UNDER THAT DATE.              */      00003710
                                                                          00003720
                                                                          00003730
```

```
                IF P(I) = PUB THEN DO;                                  00003740
                                   J = J + 1;                           00003750
                                   CNT(I) = J;                          00003760
                                   V(I,J) = VT;                         00003770
                                   FV(I,J) = FVT;                       00003780
                                   END;                                 00003790
                                                                        00003800
                                                                        00003810
              /* IF THE DATE IS NEW, THIS DOCUMENT TAKES THE FIRST       00003820
                 POSITION UNDER THE NEW DATE, AND THE NEW DATE          00003830
                 TAKES ITS PLACE IN THE LINE OF DATES.          */      00003840
                                                                        00003850
                                                                        00003860
                               ELSE DO;                                 00003870
                                   I = I + 1;                           00003880
                                   P(I) = PUB;                          00003890
                                   J = 1;                               00003900
                                   V(I,1) = VT;                         00003910
                                   FV(I,J) = FVT;                       00003920
                                   CNT(I) = 1;                          00003930
                                   END;                                 00003940
                                                                        00003950
                                                                        00003960
    /*   CONTINUE READING -SORTOUT-.                                   */00003970
                                                                        00003980
                                                                        00003990
                GO TO RD_ON;                                            00004000
                END;                                                    00004010
                                                                        00004020
                                                                        00004030
    /*   WITH ALL THE DATA FOR A SUB-CODE, WE CAN NOW PUT THIS BLOCK IN  00004040
         THE PRINTOUT.                                                 */00004050
                                                                        00004060
                                                                        00004070
           ELSE DO;                                                     00004080
                                                                        00004090
                                                                        00004100
    /*   CHECK THE NUMBER OF DATES < 15 FOR THIS SUB-CODE.  BREAK       00004110
         UP THE SUB-CODE NAME IF IT IS TOO WIDE FOR THE COLUMN.        */00004120
                                                                        00004130
                                                                        00004140
                IF I>15 THEN PUT EDIT ('***MORE THAN 15 DATES UNDER-- ',00004150
                  PREV,' ***') (SKIP(2),A(30),A(132),A(4));             00004160
                L = LENGTH(PREV);                                       00004170
                IF L>WIDTH THEN CALL BREAK(PREV,WIDTH,PC,BLK);          00004180
                           ELSE DO;                                     00004190
                                BLK = 1;                                00004200
                                PC(1) = PREV;                           00004210
                                END;                                    00004220
                                                                        00004230
                                                                        00004240
    /*   WE ADD THE DATA TO THE INDEX, GOING BY ONE DATE AT A TIME.  WE 00004250
         FILL -TEMP_OUT- WITH THE DATE AND ALL THE VT NUMBERS UNDER IT. 00004260
         FIRST, PUT THE DATE P(J) INTO -TEMP_OUT-.                     */00004270
                                                                        00004280
                                                                        00004290
                LOOP1: DO J=1 TO I;                                     00004300
                       IF CNT(J)>200 THEN PUT EDIT                      00004310
                         ('***MORE THAN 200 DOCUMENTS UNDER ',P(J),     00004320
                         ' FOR KEYWORD ',PC(1),' ***')                  00004330
                         (SKIP(2),A(33),A(2),A(13),A(WIDTH),A(4));      00004340
                       KNT = 1;                                         00004350
```

```
                    TEMP_OUT(I) = P(J) || ': ';                          00004360
                    LNE = 0;                                             00004370
                    MT = FV(J,1)/1000;                                   00004380
                                                                         00004390
                                                                         00004400
                  /* ADD THE VT NUMBERS UNDER DATE -P(J)- INTO           00004410
                     -TEMP_OUT-, BUT BREAK THEM UP BY THOUSAND           00004420
                     INTERVALS AS YOU DO THIS.  -KNT- IS THE             00004430
                     NUMBER OF ELEMENTS IN -TEMP_OUT- THIS TOOK. */      00004440
                                                                         00004450
                                                                         00004460
                    LOOP2: DO L = 1 TO CNT(J);                           00004470
                           IF MT¬=(FV(J,L)/1000) THEN DO;                00004480
                                                 MT=FV(J,L)/1000;        00004490
                                                 LNE=0;                  00004500
                                                 KNT=KNT+1;              00004510
                                                 TEMP_OUT(KNT)=          00004520
                                                   '    ';              00004530
                                                 END;                    00004540
                           IF (L¬=1) & (LNE¬=0) THEN TEMP_OUT(KNT) =      00004550
                             TEMP_OUT(KNT) || ',';                       00004560
                           TEMP_OUT(KNT)=TEMP_OUT(KNT)||' '||V(J,L);     00004570
                           LNE = LNE + 1;                                00004580
                           IF (LNE = LIM)                                00004590
                            &(FV(J,L+1)/1000=MT)                         00004600
                            &(L ¬= CNT(J))                               00004610
                                       THEN DO;                          00004620
                                            TEMP_OUT(KNT)=TEMP_OUT(KNT00004630
                                              ) || ',';                  00004640
                                            KNT=KNT+1;                   00004650
                                            LNE = 0;                     00004660
                                            TEMP_OUT(KNT)='    ';        00004670
                                            END;                         00004680
                           END LOOP2;                                    00004690
                                                                         00004700
                                                                         00004710
  /*   START ADDING THIS BLOCK OF DATA TO THE COLUMN ARRANGEMENT.  IF     00004720
       THIS IS THE FIRST DATE FOR THE SUB-CODE, PUT THE SUB-CODE         00004730
       NAME INTO THE COLUMN.                                          */00004740
                                                                         00004750
                                                                         00004760
                    IF J ¬= 1 THEN GO TO LP2;                            00004770
                    IF (LN+BLK+2) > SIZE THEN CALL HEADR(0);             00004780
                    NN = NN + 1;                                         00004790
                    LN = LN + 1;                                         00004800
                    LP1: DO L = 1 TO BLK;                                00004810
                         COL#(IND,NN) = PC(L);                           00004820
                         NN = NN + 1;                                    00004830
                         END LP1;                                        00004840
                    LN = LN + BLK;                                       00004850
                                                                         00004860
                                                                         00004870
                  /* START ADDING THE VT NUMBERS TO THE COLUMN.          00004880
                     IF WE FINISH FILLING THE COLUMN, A NEW ONE          00004890
                     MUST BE STARTED.  THIS CAUSES THE SUB-CODE          00004900
                     NAME AND SOMETIMES THE DATE TO BE REPEATED.         00004910
                     CONTINUE UNTIL THE WHOLE BLOCK IS IN                00004920
                     COLUMNS.                                     */     00004930
                                                                         00004940
                                                                         00004950
                    LP2: DO L = 1 TO KNT;                                00004960
                         IF (LN + 1) > SIZE                              00004970
```

```
                                THEN DO;                                        00004980
                                     CALL HEADR(1);                             00004990
                                     DO N = 1 TO BLK;                           00005000
                                     COL#(IND,NN) = PC(N);                      00005010
                                     NN = NN + 1;                               00005020
                                     LN = LN + 1;                               00005030
                                     END;                                       00005040
                                     IF LENGTH(PC(BLK))+12 > WIDTH              00005050
                                       THEN DO;                                 00005060
                                            COL#(IND,NN)='(CONTINUED)';         00005070
                                            NN = NN + 1;                        00005080
                                            LN = LN + 1;                        00005090
                                            END;                                00005100
                                       ELSE COL#(IND,NN-1)= PC(BLK) ||          00005110
                                                ' (CONTINUED)';                 00005120
                                     IF L ¬= 1 THEN TEMP_OUT(L)= P(J) ||        00005130
                                                ': ' || SUBSTR(TEMP_OUT(L),5);  00005140
                                     END;                                       00005150
                                COL#(IND,NN) = TEMP_OUT(L);                     00005160
                                NN = NN + 1;                                    00005170
                                LN = LN + 1;                                    00005180
                                END LP2;                                        00005190
                           END LOOP1;                                           00005200
                                                                                00005210
                                                                                00005220
   /*   IF THERE IS MORE DATA, START THIS PROCEDURE FOR THE NEXT SUB-CODE.      00005230
        OTHERWISE, PUT THE PAGE BEING FILLED ON TO THE OUTPUT FILE.     */      00005240
                                                                                00005250
                                                                                00005260
                     IF IMP = 0 THEN GO TO INITL;                               00005270
                     END;                                                       00005280
   LOOP4:  DO K = 1 TO SIZE-LINES2;                                             00005290
           PUT FILE (OUTS) SKIP(1);                                             00005300
   LOOP5:  DO L = 1 TO COL;                                                     00005310
           PUT FILE (OUTS) EDIT (COL#(L,K))                                     00005320
             (SKIP(0),COLUMN(MG(L)),A(WIDTH));                                  00005330
           END LOOP5;                                                           00005340
           END LOOP4;                                                           00005350
   STOP:   PUT FILE (OUTS) PAGE;                                                00005360
           PUT DATA (PAGE);                                                     00005370
           CLOSE FILE (SORTOUT),                                                00005380
                 FILE (OUTS);                                                   00005390
                                                                                00005400
                                                                                00005410
   /*   THIS PROCEDURE STARTS A NEW COLUMN.                             */      00005420
                                                                                00005430
                                                                                00005440
   HEADR:  PROCEDURE (NP);                                                      00005450
           DECLARE                                                              00005460
             (K,L)              FIXED BINARY (31,0),                            00005470
             NP                 FIXED BINARY (31,0),                            00005480
           ;                                                                    00005490
                                                                                00005500
                                                                                00005510
   /*   UP THE COLUMN SUBSCRIPT FOR THE NEXT COLUMN.                    */      00005520
                                                                                00005530
                                                                                00005540
           IND = IND + 1;                                                       00005550
                                                                                00005560
                                                                                00005570
   /*   IF ALL THE COLUMNS ON THE PAGE ARE FULL, WE MUST PUT THE COLUMNS       00005580
        ON THE OUTPUT FILE.  THEN A TITLE IS ADDED TO THE OUTPUT FILE          00005590
```

```
      FOR A NEW PAGE.  WE WILL BE FILLING COLUMN 1 OF THE NEW PAGE.    */00005600
                                                                        00005610
                                                                        00005620
         IF IND = IMAX                                                  00005630
           THEN DO;                                                     00005640
                IND = 1;                                                00005650
                HD1: DO L = 1 TO SIZE-LINES2;                           00005660
                     PUT FILE (OUTS) SKIP(1);                           00005670
                HD2: DO K = 1 TO COL;                                   00005680
                     PUT FILE (OUTS) EDIT (COL#(K,L))                   00005690
                       (SKIP(0),COLUMN(MG(K)),A(WIDTH));                00005700
                     COL#(K,L) = ' ';                                   00005710
                     END HD2;                                           00005720
                     END HD1;                                           00005730
                PAGE = PAGE + 1;                                        00005740
                IF SKPP = MARG THEN DO;                                 00005750
                                    SKPP = SZ - 3;                      00005760
                                    SKPH = MARG;                        00005770
                                    END;                                00005780
                               ELSE DO;                                 00005790
                                    SKPP = MARG;                        00005800
                                    SKPH = SZ - 12;                     00005810
                                    END;                                00005820
                PUT FILE (OUTS) EDIT (PAGE,HEAD,TITLE)                  00005830
                  (PAGE,LINE(LINEH2),COLUMN(SKPP),F(4),SKIP(0),COLUMN(  00005840
                  SKPH),A(13),LINE(LINET),X((SZ-LTH-MARG+1)/2+MARG-1),  00005850
                  A(LTH));                                              00005860
                PUT FILE (OUTS) SKIP(LINES2-LINET);                     00005870
                END;                                                    00005880
                                                                        00005890
                                                                        00005900
   /*   INITIALIZE LINE COUNTERS FOR THE NEW COLUMN.                   */00005910
                                                                        00005920
                                                                        00005930
         IF FRST = PAGE THEN LN = LINES1;                               00005940
                        ELSE LN = LINES2;                               00005950
         NN = NP;                                                       00005960
         IF NN = 0 THEN LN = LN - 1;                                    00005970
         RETURN;                                                        00005980
         END HEADR;                                                     00005990
                                                                        00006000
                                                                        00006010
   /*   THIS PROCEDURE TAKES A STRING -BREAKE- AND BREAKS IT UP INTO    00006020
        PIECES OF LENGTH -LNTH-, STORING THEM IN -BROKE-.  -NU- IS THE  00006030
        NUMBER OF PIECES IT TOOK.                                       */00006040
                                                                        00006050
                                                                        00006060
   BREAK: PROCEDURE(BREAKE,LNTH,BROKE,NU);                              00006070
         DECLARE                                                        00006080
           BREAKE              CHARACTER (132) VARYING,                 00006090
           LNTH                FIXED BINARY (31,0),                     00006100
           BROKE(4)            CHARACTER (WIDTH) VARYING,               00006110
           NU                  FIXED BINARY (31,0),                     00006120
           L                   FIXED BINARY (31,0),                     00006130
           SUM1                FIXED BINARY (31,0),                     00006140
           TEMP                CHARACTER (132) VARYING,                 00006150
         ;                                                              00006160
         NU = 0;                                                        00006170
   INIT: SUM1 = 0;                                                      00006180
         TEMP = BREAKE;                                                 00006190
   SRCH: L = INDEX(TEMP,' ');                                           00006200
         IF L = 0                                                       00006210
```

```
            THEN DO;                                                    00006220
                 L = LENGTH(TEMP);                                      00006230
                 SUM1 = SUM1 + L;                                       00006240
                 IF SUM1 > LNTH                                         00006250
                   THEN DO;                                             00006260
                        NU = NU + 1;                                    00006270
                        BROKE(NU) = SUBSTR(BREAKE,1,SUM1-L-1);          00006280
                        NU = NU + 1;                                    00006290
                        BROKE(NU) = SUBSTR(BREAKE,SUM1-L+1);            00006300
                        RETURN;                                         00006310
                        END;                                            00006320
                   ELSE DO;                                             00006330
                        NU = NU + 1;                                    00006340
                        BROKE(NU) = SUBSTR(BREAKE,1,SUM1);              00006350
                        RETURN;                                         00006360
                        END;                                            00006370
                 END;                                                   00006380
SUM:        SUM1 = SUM1 + (L - 1);                                      00006390
            IF SUM1 <=LNTH THEN DO;                                     00006400
                                TEMP = SUBSTR(TEMP,L+1);                00006410
                                SUM1 = SUM1 + 1;                        00006420
                                GO TO SRCH;                             00006430
                                END;                                    00006440
            NU = NU + 1;                                                00006450
            BROKE(NU) = SUBSTR(BREAKE,1,SUM1-L);                        00006460
            BREAKE = SUBSTR(BREAKE,SUM1+2-L);                           00006470
            SUM1 = 0;                                                   00006480
            GO TO SUM;                                                  00006490
            END BREAK;                                                  00006500
                                                                        00006510
                                                                        00006520
/*    THIS PROCEDURE TAKES A VT NUMBER LIKE '003406' AND ELIMINATES     00006530
      LEADING ZEROES SO IT BECOMES '3406'.                            */00006540
                                                                        00006550
                                                                        00006560
DELET:                                                                  00006570
            PROCEDURE;                                                  00006580
            DECLARE                                                     00006590
              M                 FIXED BINARY (31,0),                    00006600
              TEMP              CHARACTER (6) VARYING,                  00006610
              BLANK             CHARACTER (1) INITIAL (' '),            00006620
            ;                                                           00006630
            TEMP = VT;                                                  00006640
LP:         DO M = 1 TO 6;                                              00006650
            IF SUBSTR(TEMP,M,1)='0' THEN DO;                            00006660
                                  IF M=1 THEN VT=BLANK||SUBSTR(VT,2);   00006670
                                         ELSE VT=SUBSTR(VT,1,M-1) ||    00006680
                                                 BLANK            ||    00006690
                                                 SUBSTR(VT,M+1);        00006700
                                END;                                    00006710
                              ELSE GO TO FIX;                           00006720
            END LP;                                                     00006730
FIX:        VT = SUBSTR(VT,6-WD+1,WD);                                  00006740
            RETURN;                                                     00006750
            END DELET;                                                  00006760
                                                                        00006770
            END BUSINESS;                                               00006780
                                                                        00006790
            END ANNVCT;                                                 00006800
```

DOCUMENT NUMBER ANNUAL

```
/*****************************************************************/00000010
/*                                                               */00000020
/*                                                               */00000030
/*      A N N U A L   O O C U M E N T   N U M B E R   I N D E X  */00000040
/*                          P R O G R A M                        */00000050
/*                                                               */00000060
/*                                                               */00000070
/*  PURPOSE:  THIS PROGRAM PROOUCES THE ANNUAL OOCUMENT NUMBER INOEX.*/00000080
/*            IT TAKES THE FOUR ARM TAPE DATA SETS AND THE FOUR AIM */00000090
/*            DATA SETS AND CONCATENATES ALL 8 TOGETHER.  THEN HE   */00000100
/*            SORT THE RECORDS AND PRINT THEM IN A FORMAT SPECIFIED */00000110
/*            BY THE VOCATIONAL CENTER.  THIS GIVES US AN ANNUAL    */00000120
/*            INOEX.                                               */00000130
/*                                                                 */00000140
/*  DEFINITION OF VARIABLES:                                       */00000150
/*     1. COL -- NUMBER OF COLUMNS IN THE INDEX PRINTOUT.          */00000160
/*     2. HEAD -- HEADING AT THE TOP OF EVERY INDEX PAGE WHICH HAS */00000170
/*                THIS FORM:  'AIM/ARM 69-70'.                     */00000180
/*     3. LINEH1 -- THE LINE ON THE TITLE PAGE ON WHICH -HEAO- AND */00000190
/*                  THE PAGE NUMBER APPEAR.                        */00000200
/*     4. LINEH2 -- SAME AS  -LINEH1- BUT FOR ALL SUCCEEOING PAGES. */00000210
/*     5. LINES1 -- THE LINE ON THE TITLE PAGE ON WHICH THE SUB-TITLE*/00000220
/*                  APPEARS.                                       */00000230
/*     6. LINES2 -- SAME AS -LINES1- BUT FOR ALL SUCCEEOING PAGES. */00000240
/*     7. LINET -- LINE ON ALL PAGE BUT THE TITLE PAGE ON WHICH    */00000250
/*                 THE INDEX TITLE APPEARS.                        */00000260
/*     8. LT -- LENGTH OF INDEX TITLE.                             */00000270
/*     9. MARG -- LEFT-HAND MARGIN OF THE PAGE.                    */00000280
/*    10. MG -- ARRAY OF COLUMN MARGINS, EACH MEASUREO FROM COLUMN */00000290
/*              1.                                                 */00000300
/*    11. NBH -- NUMBER OF BLOCKS OF OUTPUT ON THE TITLE PAGE.  A  */00000310
/*               BLOCK OF OUTPUT IS A FEW LINES OF OUTPUT, ANO     */00000320
/*               BLOCKS ARE SEPARATED BY A BLANK LINE.             */00000330
/*    12. NBR -- NUMBER OF BLOCKS OF OUTPUT ON ALL OTHER PAGES.    */00000340
/*    13. NPB -- NUMBER OF LINES OF OUTPUT IN ONE BLOCK.           */00000350
/*    14. PAGE -- PAGE NUMBER OF AN INDEX PAGE.                    */00000360
/*    15. QUE -- INOICATES WHETHER THE TITLE PAGE WILL BE ON THE LEFT*/00000370
/*               (L) OR RIGHT (R) SIOE OF THE PUBLICATION.         */00000380
/*    16. SIZE -- MAXIMUM PAGE LENGTH ALLOWEO.  THIS INCLUOES HEADING*/00000390
/*                ANO INDEX ON THE PAGE.                           */00000400
/*    17. SORTIN -- EIGHT SUBSET DATA SETS FROM QUARTERLY RUNS, ALL*/00000410
/*                  CONCATENATEO TOGETHER TO FORM ONE LARGE SET.   */00000420
/*    18. SORTOUT -- SORTED DISK FILE.                             */00000430
/*    19. SPACE1 -- NUMBER OF COLUMNS BETWEEN THE VT NUMBER AND    */00000440
/*                  ISSUE IN THE PRINTOUT.                         */00000450
/*    20. SPACE2 -- NUMBER OF COLUMNS BETWEEN THE ISSUE AND PAGE   */00000460
/*                  NUMBER IN THE PRINTOUT.                        */00000470
/*    21. STR -- STRING USED TO INPUT AND OUTPUT FILE RECORDS.     */00000480
/*    22. SZ -- RIGHT-HAND MARGIN OF THE PAGE.                     */00000490
/*    23. TITL -- TITLE OF THE INDEX.                              */00000500
/*    24. WO -- NUMBER OF SIGNIFICANT DIGITS IN THE VT NUMBER.     */00000510
/*    25. COL# -- REPRESENTS THE COLUMNS OF A PAGE STARTING BELOW  */00000520
/*                THE HEADING.  THE FIRST SUBSCRIPT OF THIS MATRIX */00000530
/*                REFERENCES A COLUMN AND THE SECOND SUBSCRIPT     */00000540
/*                REFERENCES A LINE IN THAT COLUMN.                */00000550
/*    26. IND -- THE CURRENT COLUMN BEING FILLED ON AN INDEX PAGE. */00000560
/*    27. ISSUE -- IS A 3 CHARACTER CODE; THE FIRST CHARACTER IS 'A' */00000570
/*                 FOR 'AIM' OR ' ' FOR 'ARM'; THE SECONO ANO THIRD */00000580
/*                 CHARACTERS GIVE THE SEASON 'F ','W ','SP','SU'. */00000590
/*    28. NN -- THE NEXT LINE TO BE FILLEO IN A COLUMN OF AN INOEX */00000600
/*              PAGE.                                              */00000610
/*    29. PG -- A PAGE NUMBER READ OFF A SUBSET FILE RECORD.       */00000620
```

```
/*     30. SKPH -- COLUMN IN WHICH WE START PRINTING -HEAD-.                  */00000630
/*     31. SKPP -- COLUMN IN WHICH WE START PRINTING THE PAGE NUMBER          */00000640
/*                 OF AN INDEX PAGE.                                          */00000650
/*     32. VT -- A VT NUMBER READ OFF OF A SUBSET FILE RECORD.                */00000660
/*     33. YR -- THE YEAR OF THE 'AIM' OR 'ARM' ISSUE WHEN A DOCUMENT         */00000670
/*                APPEARED IN IT.                                             */00000680
/*     34. IHESRTA -- PL/1 CALL TO SORT.                                      */00000690
/*     35. RETCDE -- CODE RETURNED BY THE SORT INDICATING SUCCESS OR          */00000700
/*                   FAILURE.                                                 */00000710
/*     36. SFIELD -- INDICATES TO SORT WHAT PART OF THE RECORDS TO            */00000720
/*                   SORT UNDER.                                              */00000730
/*     37. RFIELD -- DEFINES FOR SORT THE TYPE OF RECORDS TO BE               */00000740
/*                   SORTED.                                                  */00000750
/*                                                                            */00000760
/*  INPUT TO THE PROGRAM:                                                     */00000770
/*                                                                            */00000780
/*     -SORTIN-                                                               */00000790
/*       THIS IS THE 8 DATA SETS, ON ONE OR TWO MAGNETIC TAPES, WHICH         */00000800
/*       ARE SAVED AFTER EVERY QUARTERLY RUN OF AIM AND ARM.                  */00000810
/*     -SYSIN-, INPUT ON CARDS                                                */00000820
/*       THE INPUT ON CARDS RELAYS THE OUTPUT SPECIFICATIONS TO THE           */00000830
/*       PROGRAM.  THE DATA IS ONE CONTINUOUS STREAM ON THE CARDS             */00000840
/*       AND IT WILL TAKE SEVERAL CARDS.  THE VALUES OF THE VARIABLES         */00000850
/*       IN THEIR ORDER BELOW ARE THE DATA ON THE CARDS.                      */00000860
/*            -PAGE-          F(4)                                            */00000870
/*            -SIZE-          F(4)                                            */00000880
/*            -WD-            F(4)                                            */00000890
/*            -MARG-          F(4)                                            */00000900
/*            -SZ-            F(4)                                            */00000910
/*            -LINEH1-        F(4)                                            */00000920
/*            -LINES1-        F(4)                                            */00000930
/*            -LINEH2-        F(4)                                            */00000940
/*            -LINET-         F(4)                                            */00000950
/*            -LINES2-        F(4)                                            */00000960
/*            -SPACE1-        F(4)                                            */00000970
/*            -SPACE2-        F(4)                                            */00000980
/*            -COL-           F(4)                                            */00000990
/*            -MG- ARRAY      COL * F(4)                                      */00001000
/*            -NPB-           F(4)                                            */00001020
/*            -NBR-           F(4)                                            */00001030
/*            -LT-            F(4)                                            */00001040
/*            -TITL-          A(LT)                                           */00001050
/*            -HEAD-          A(13)                                           */00001060
/*            -QUE-           A(1)                                            */00001070
/*       THESE VARIABLES WERE DEFINED IN THE PREVIOUS SECTION AND             */00001080
/*       THEIR VALUES CAN BE OBTAINED FROM THE TYPED WRITE-UP OR THE          */00001090
/*       VOCATIONAL CENTER.                                                   */00001100
/*                                                                            */00001110
/*  SPECIAL NOTE:                                                             */00001120
/*     PRINT THE INDEX ON THE PRINTER UNTIL IT HAS BEEN APPROVED BY           */00001130
/*     THE VOCATIONAL CENTER.  THEN IT CAN BE PRINTED ON TAPE AND THE         */00001140
/*     TAPE CAN BE LISTED ON THE 360/50 WITH THE TN CHAIN.  THIS WILL         */00001150
/*     GIVES US AN UPPER AND LOWER CASE LISTING.                              */00001160
/*                                                                            */
/******************************************************************************/00001170
                                                                               00001180
ANNDOC: PROCEDURE OPTIONS (MAIN);                                              00001190
        DECLARE                                                                00001200
          COL                   FIXED BINARY (31,0),                           00001210
          HEAD                  CHARACTER (13),                                00001220
          I                     FIXED BINARY (31,0),                           00001230
          LINEH1                FIXED BINARY (31,0),                           00001240
          LINEH2                FIXED BINARY (31,0),                           00001250
```

```
          LINES1              FIXED BINARY (31,0),                          00001260
          LINES2              FIXED BINARY (31,0),                          00001270
          LINET               FIXED BINARY (31,0),                          00001280
          LT                  FIXED BINARY (31,0),                          00001290
          MARG                FIXED BINARY (31,0),                          00001300
          MG(4)               FIXED BINARY (31,0),                          00001310
          NBH                 FIXED BINARY (31,0),                          00001320
          NBR                 FIXED BINARY (31,0),                          00001330
          NPB                 FIXED BINARY (31,0),                          00001340
          PAGE                FIXED BINARY (31,0),                          00001350
          QUE                 CHARACTER (1),                                00001360
          SIZE                FIXED BINARY (31,0),                          00001370
          SORTIN              FILE SEQUENTIAL RECORD,                       00001380
          SORTOUT             FILE SEQUENTIAL RECORD,                       00001390
          SPACE1              FIXED BINARY (31,0),                          00001400
          SPACE2              FIXED BINARY (31,0),                          00001410
          STR                 CHARACTER (18),                               00001420
          SZ                  FIXED BINARY (31,0),                          00001430
          TITL                CHARACTER  50) VARYING,                       00001440
          WD                  FIXED BINARY (31,0),                          00001450
        ;                                                                   00001460
                                                                            00001470
                                                                            00001480
   /* SORT THE 8 SUBSET FILES BY VT NUMBER IN ASCENDING ORDER.  PRINT      */00001490
   /* OUT THE RETURN CODE.                                                 */00001500
                                                                            00001510
                                                                            00001520
   SORT:  BEGIN;                                                            00001530
          DECLARE                                                           00001540
            IHESRTA           ENTRY(,,FIXED BINARY (31,0),),                00001550
            RETCDE            FIXED BINARY (15,0),                          00001560
            RFIELD            CHARACTER (27),                               00001570
            SFIELD            CHARACTER (35),                               00001580
          ;                                                                 00001590
          SFIELD = ' SORT FIELDS=(1,6,CH,A),SIZE=E9999 ';                   00001600
          RFIELD = ' RECORD TYPE=F,LENGTH=(18) ';                           00001610
          CALL IHESRTA (SFIELD,RFIELD,45056,RETCDE);                        00001620
          PUT EDIT (RETCDE)  (SKIP(3),F(3));                                00001630
          END SORT;                                                         00001640
                                                                            00001650
                                                                            00001660
   /* PRINT OUT THE SORTED RECORDS AS A CHECK.                             */00001670
                                                                            00001680
                                                                            00001690
          ON ENDFILE (SORTOUT) GO TO CLOSE;                                 00001700
          OPEN FILE (SORTOUT);                                              00001710
   RD:    READ FILE (SORTOUT) INTO (STR);                                   00001720
            PUT EDIT(STR)(X(5),A(18));                                      00001730
          GO TO RD;                                                         00001740
   CLOSE: CLOSE FILE (SORTOUT);                                             00001750
                                                                            00001760
                                                                            00001770
   /* READ IN ALL OUTPUT SPECIFICATIONS NEEDED TO PRINT THE INDEX.         */00001780
                                                                            00001790
                                                                            00001800
          GET EDIT (PAGE,SIZE,WD,MARG,SZ,LINEH1,LINES1,LINEH2,LINET,        00001810
            LINES2,SPACE1,SPACE2,COL,(MG(I) DO I = 1 TO COL),NPB,NBH,NBR,   00001820
            LT,TITL,HEAD,QUE)                                               00001830
            (17 F(4),(COL) F(4),A(LT),A(13),A(1));                          00001840
                                                                            00001850
                                                                            00001860
   /* PRINTING OCCURS IN THIS BLOCK.                                       */00001870
```

```
                                                                        00001880
                                                                        00001890
PRINT:  BEGIN;                                                          00001900
        DECLARE                                                         00001910
          COL#(COL,SIZE-LINES2-1) CHARACTER (SPACE1+15+SPACE2),         00001920
          I                       FIXED BINARY (31,0),                  00001930
          IND                     FIXED BINARY (31,0),                  00001940
            ISSUE                 CHARACTER (3),                        00001950
          J                       FIXED BINARY (31,0),                  00001960
          L                       FIXED BINARY (31,0),                  00001970
          NN                      FIXED BINARY (31,0),                  00001980
          PG                      CHARACTER (4),                        00001990
          SKPH                    FIXED BINARY (31,0),                  00002000
          SKPP                    FIXED BINARY (31,0),                  00002010
          VT                      CHARACTER (6) VARYING,                00002020
          YR                      CHARACTER (2),                        00002030
        ;                                                               00002040
                                                                        00002050
        ON ENDFILE (SORTOUT) GO TO STOP;                                00002060
        OPEN FILE (OUTS) STREAM PRINT OUTPUT PAGESIZE(SIZE-4);          00002070
        OPEN FILE (SORTOUT);                                            00002080
                                                                        00002090
                                                                        00002100
  /* MAKE INITIALIZATIONS TO BEGIN PRINTING.                           */00002110
                                                                        00002120
                                                                        00002130
        IF QUE = 'R' THEN DO;                                           00002140
                          SKPP = SZ - 3;                                00002150
                          SKPH = MARG;                                  00002160
                          END;                                          00002170
                     ELSE DO;                                           00002180
                          SKPP = MARG;                                  00002190
                          SKPH = SZ - 12;                               00002200
                          END;                                          00002210
BLKOT:  DO I = 1 TO SIZE-LINES2-1;                                      00002220
        DO J = 1 TO COL;                                                00002230
        COL#(J,I) = ' ';                                                00002240
        END;                                                            00002250
        END BLKOT;                                                      00002260
        L = NBH;                                                        00002270
        NN = 1;                                                         00002280
        IND = 1;                                                        00002290
                                                                        00002300
                                                                        00002310
  /* PUT A HEADING ON THE TITLE PAGE:  PAGE NUMBER, 'AIM/ARM YR1-YR2',  */00002320
  /* AND SUB-TITLE.                                                    */00002330
                                                                        00002340
                                                                        00002350
        PUT FILE (OUTS) EDIT (PAGE,HEAD)                                00002360
          (PAGE,LINE(LINEH1),COLUMN(SKPP),F(4),SKIP(0),COLUMN(SKPH),    00002370
          A(13));                                                       00002380
        PUT FILE (OUTS) LINE(LINES1);                                   00002390
LP1:    DO I = 1 TO COL;                                                00002400
        PUT FILE (OUTS) EDIT ('VT NO','ISSUE','PAGE')                   00002410
          (SKIP(0),COLUMN(MG(I)),A(5),X(SPACE1),A(5),X(SPACE2),A(4));   00002420
        END LP1;                                                        00002430
        PUT FILE (OUTS) SKIP(1);                                        00002440
                                                                        00002450
                                                                        00002460
  /* WE NOW FILL UP A COLUMN OF AN INDEX PAGE WITH INDEX DATA.  THE     */00002470
  /* COLUMN HAS A FEW LINES OF PRINT, THEN A BLANK LINE, A FEW MORE    */00002480
  /* LINES OF PRINT, A BLANK LINE, ETC.                                */00002490
```

```
                                                                        00002500
                                                                        00002510
LP2:    DO I = 1 TO L;                                                  00002520
LP3:    DO J = 1 TO NPB;                                                00002530
        READ FILE (SORTOUT) INTO (STR);                                 00002540
        GET STRING (STR) EDIT (VT,PG,ISSUE,YR)                          00002550
          (A(6),A(4),A(3),A(2));                                        00002560
        CALL DELETE;                                                    00002570
        PUT STRING (COL#(IND,NN)) EDIT (VT,ISSUE,YR,PG)                 00002580
          (X(6-MD),A(MD),X(SPACE1),A(3),A(2 ,X(SPACE2),A(4));           00002590
        NN = NN + 1;                                                    00002600
        END LP3;                                                        00002610
        NN = NN + 1;                                                    00002620
        END LP2;                                                        00002630
                                                                        00002640
                                                                        00002650
    /* NOW WE MUST START A NEW COLUMN TO BE FILLED ON THIS PAGE.  SO    */00002660
    /* WE UP THE COLUMN SUBSCRIPT.  IF WE HAVE FILLED THE LAST COLUMN   */00002670
    /* ON THIS PAGE, CONTINUE HERE.                                     */00002680
                                                                        00002690
                                                                        00002700
        IND = IND + 1;                                                  00002710
        IF IND = (COL + 1)                                              00002720
                                                                        00002730
                                                                        00002740
                /* ADD THE FINISHED PAGE TO THE OUTPUT    */            00002750
                /* FILE.                                  */            00002760
                                                                        00002770
                                                                        00002780
          THEN DO;                                                      00002790
               IND = 1;                                                 00002800
               LP4: DO I = 1 TO L*NPB+L-1;                              00002810
                    PUT FILE (OUTS) SKIP(1);                            00002820
               LP5: DO J = 1 TO COL;                                    00002830
                    PUT FILE (OUTS) EDIT (COL#(J,I))                    00002840
                      (SKIP(0),COLUMN(MG(J)-1),A(15+SPACE1+SPACE2));    00002850
                    COL#(J,I) = ' ';                                    00002860
                    END LP5;                                            00002870
                    END LP4;                                            00002880
                                                                        00002890
                                                                        00002900
                /* PUT A NEW HEADING ON THE OUTPUT FILE   */            00002910
                /* FOR THE NEW PAGE.                      */            00002920
                                                                        00002930
                                                                        00002940
               L = NBR;                                                 00002950
               PAGE = PAGE + 1;                                         00002960
               IF SKPP = MARG THEN DO;                                  00002970
                                   SKPP = SZ - 3;                       00002980
                                   SKPH = MARG;                         00002990
                                   END;                                 00003000
                              ELSE DO;                                  00003010
                                   SKPP = MARG;                         00003020
                                   SKPH = SZ - 12;                      00003030
                                   END;                                 00003040
               PUT FILE (OUTS) EDIT (PAGE,HEAD,TITL)                    00003050
                 (PAGE,LINE(LINEH2),COLUMN(SKPP),F(4),SKIP(0),COLUMN(   00003060
                 SKPH),A(13),LINE(LINET),X((SZ-MARG+1-LT)/2+MARG-1),    00003070
                 A(LT));                                                00003080
               PUT FILE (OUTS) LINE(LINES2);                            00003090
               LP6: DO I = 1 TO COL;                                    00003100
                    PUT FILE (OUTS) EDIT ('VT NO','ISSUE','PAGE')       00003110
```

```
                    (SKIP(0),COLUMN(MG(I)),A(5),X(SPACE1),A(5),
                    X(SPACE2),A(4));
                END LP6;
            PUT FILE (OUTS) SKIP(1);
            END;


/* NEW PAGE OR NOT, INITIALIZE THE LINE COUNTER AND CONTINUE
/* PROCESSING THE INDEX DATA.

        NN = 1;
        GO TO LP2;


/* THIS PROCEDURE DELETES LEADING ZEROES FROM A VT NUMBER.  THUS,
/* '007095' BECOMES '7095'.

DELETE: PROCEDURE;
          DECLARE
          M                 FIXED BINARY (31,0),
          TEMP              CHARACTER (6) VARYING,
        ;
        TEMP = VT;
LP8:    DO M = 1 TO 6;
        IF SUBSTR(TEMP,M,1) = '0'
          THEN DO;
               IF M = 1 THEN VT = ' ' || SUBSTR(VT,2);
                        ELSE VT = SUBSTR(VT,1,M-1) || ' ' || SUBSTR(VT,
                                                              M+1);
               END;
               ELSE GO TO FIX;
        END LP8;
FIX:    VT = SUBSTR(VT,6-WD+1,WD);
        RETURN;
        END DELETE;


/* WHEN ALL THE DATA HAS BEEN READ OFF THE DISK FILE, WE ARE ALMOST
/* THROUGH.  ADD THE PAGE THAT WAS BEING FILLED TO THE OUTPUT FILE.
/* CLOSE ALL FILES AND TERMINATE THE PROGRAM.  THE INDEX IS COMPLETE.


STOP:     CLOSE FILE (SORTOUT);
          LP7: DO I = 1 TO L*NPB+L-1;
               PUT FILE (OUTS) SKIP(1);
          LP8: DO J = 1 TO COL;
               PUT FILE (OUTS) EDIT (COL#(J,I))
                 (SKIP(0),COLUMN(MG(J)-1),A(15+SPACE1+SPACE2));
               END LP8;
               END LP7;
          PUT FILE (OUTS) PAGE;
          CLOSE FILE (OUTS);
          PUT EDIT ('LAST PAGE USED IS ',PAGE) (SKIP(3),A(18),F(4));

        END PRINT;

        END ANNDOC;
```

EDIT OF SUBJECT, AUTHOR, AND
SUPPORTING SERVICES

```
/*****************************************************************************/00000010
/*                                                                           */00000020
/*    E D I T   F O R   S U B J E C T ,   A U T H O R ,   A N D              */00000030
/*              S . S .   S U B S E T   F I L E S                            */00000040
/*                                                                           */00000050
/*                                                                           */00000060
/*  PURPOSE:  THIS PROGRAM CAN COPY THE EIGHT SUBJECT DATA SETS ON           */00000070
/*            MAGNETIC TAPE TO ANOTHER TAPE.  RECORDS FROM EACH DATA         */00000080
/*            SET CAN BE ADDED, DELETED, OR REPLACED WHILE THE               */00000090
/*            COPYING IS BEING DONE.                                         */00000100
/*                                                                           */00000110
/*INPUT TO THE PROGRAM:                                                      */00000120
/*                                                                           */00000130
/*    -SYSIN-, INPUT ON CARDS:                                               */00000140
/*      THE FOLLOWING CARD SET-UP IS REPEATED FOR EACH DATA SET ON           */00000150
/*      THE TAPE.  FORMAT ON THESE CARDS IS  'FREE'; THE DATA CAN BE         */00000160
/*      IN ANY COLUMNS AS LONG AS AT LEAST ONE BLANK IS BETWEEN              */00000170
/*      VARIABLES.  THE ORDER OF THE DATA IS AS FOLLOWS:                     */00000180
/*      -M-, THE NUMBER OF RECORDS TO BE REPLACED ON THE DATA SET.           */00000190
/*      -K-, THE NUMBER OF RECORDS TO BE DELETED FROM THE DATA SET.          */00000200
/*      'REPLACEMENT DECK' -- THIS IS A DECK WITH THE DATA TO BE             */00000210
/*        REPLACED AND ITS REPLACEMENT.  FIRST, WE HAVE A KEYWORD,           */00000220
/*        A DATE, AND A VT NUMBER, EACH IN QUOTES.  THIS IS DATA IN          */00000230
/*        A RECORD WHICH IS TO BE REPLACED IN SOME PART.  THEN, WE           */00000240
/*        HAVE THE KEYWORD, DATE, AND VT NUMBER FOR THE REPLACEMENT.         */00000250
/*        THIS PATTERN CONTINUES 'M' TIMES.                                  */00000260
/*      'DELETION DECK' -- THIS GIVES THE KEYWORD, DATE, AND VT              */00000270
/*        NUMBER, EACH IN QUOTES, OF A RECORD TO BE DELETED.  THIS           */00000280
/*        OCCURRS -K- TIMES.                                                 */00000290
/*      -L-, THE NUMBER OF RECORDS TO BE ADDED TO THE DATA SET.              */00000300
/*      'ADDITION DECK' -- THIS GIVES THE KEYWORD, DATE, AND VT              */00000310
/*        NUMBER, EACH IN QUOTES, OF A RECORD TO BE ADDED.  THIS             */00000320
/*        OCCURRS -L- TIMES.                                                 */00000330
/*            NOTE:  IF L=0, THERE ARE NO ADDITIONS.  IF K=0, THERE          */00000340
/*                   ARE NO DELETIONS.  IF M=0, THERE ARE NO                 */00000350
/*                   REPLACEMENTS.  IF ALL THREE ARE ZERO, THE DATA          */00000360
/*                   SET WILL ONLY BE COPIED.                                */00000370
/*                                                                           */00000380
                                                                             00000390
EDIT:   PROCEDURE OPTIONS (MAIN);                                            00000400
DECLARE (SUB_LEN,RECORD_LEN) FIXED BINARY(31,0),                             00000410
        INDEX CHARACTER(4);                                                  00000420
        GET FILE(SYSIN) LIST(SUB_LEN,INDEX);                                 00000430
        RECORD_LEN=SUB_LEN+8;                                                00000440
    IF RECORD_LEN<18 THEN RECORD_LEN=18;                                     00000450
BEGIN;                                                                       00000460
        DECLARE                                                              00000470
          FIX                 FILE SEQUENTIAL RECORD,                        00000480
          TEMP                FILE SEQUENTIAL RECORD,                        00000490
          BLANK               CHARACTER (1),                                 00000500
          DUMMY               CHARACTER (RECORD_LEN),                        00000510
          REC(50)             CHARACTER (RECORD_LEN),                        00000520
          VT                  CHARACTER (6),                                 00C00530
          PUB                 CHARACTER(2),                                  00000540
          SUB                 CHARACTER (SUB_LEN) VARYING,                   00000550
          END                 CHARACTER (1) INITIAL ('*'),                   00000560
          SUBJ                CHARACTER (SUB_LEN) VARYING,                   00000570
          DUM(100,2)          CHARACTER (RECORD_LEN),                        00000580
        ;                                                                    00000590
  GET FILE(SYSIN) LIST (M,K);                                                00000600
        IF M = 0 THEN GO TO TEST;                                            00000610
LOOP3:  DO I = 1 TO M;                                                       00000620
```

```
LOOP3A: DO II = 1 TO 2;                                                    00000630
        GET LIST (SUB,PUB,VT);                                             00000640
                  IF INDEX='SUBJ' THEN SUBJ=SUB||END;                      00000650
                                  ELSE SUBJ=SUB;                           00000660
 DUM(I,II)=' ';                                                            00000670
PUT STRING(DUM(I,II)) EDIT(SUBJ,PUB,VT)(A(SUB_LEN),A(2),A(6));             00000680
        END LOOP3A;                                                        00000690
        END LOOP3;                                                         00000700
TEST:   IF K = 0 THEN GO TO OPEN;                                          00000710
LOOP3B: DO I = 1 TO K;                                                     00000720
        GET LIST (SUB,PUB,VT);                                             00000730
                  IF INDEX='SUBJ' THEN SUBJ=SUB||END;                      00000740
                                  ELSE SUBJ=SUB;                           00000750
    REC(I)=' ';                                                            00000760
PUT STRING(REC(I)    ) EDIT(SUBJ,PUB,VT)(A(SUB_LEN),A(2),A(6));            00000770
        END LOOP3B;                                                        00000780
OPEN: OPEN FILE(FIX) INPUT SEQUENTIAL RECORD;                              00000790
      OPEN FILE(TEMP) OUTPUT SEQUENTIAL RECORD;                            00000800
        ON ENDFILE (FIX) BEGIN;                                            00000810
                  GET LIST (L);                                            00000820
           PUT FILE (SYSPRINT) SKIP(4);                                    00000830
           IF L = 0 THEN GO TO CL;                                         00000840
           LOOP5: DO I = 1 TO L;                                           00000850
                  GET LIST (SUB,PUB,VT);                                   00000860
                  SUBJ = SUB || END;                                       00000870
    DUMMY=' ';                                                             00000880
PUT STRING(DUMMY    ) EDIT(SUBJ,PUB,VT)(A(SUB_LEN),A(2),A(6));             00000890
                  WRITE FILE (TEMP) FROM (DUMMY);                          00000900
                  PUT DATA (DUMMY);                                        00000910
                  END LOOP5;                                               00000920
 CL:    CLOSE FILE(FIX), FILE(TEMP);                                       00000930
           GO TO EL2;                                                      00000940
           END;                                                            00000950
NXT:    READ FILE (FIX) INTO (DUMMY);                                      00000960
        IF M = 0 THEN GO TO LBL1;                                          00000970
LOOP4:  DO I = 1 TO M;                                                     00000980
        IF DUMMY = DUM(I,1) THEN DO;                                       00000990
                                 DUMMY = DUM(I,2);                         00001000
                                 WRITE FILE (TEMP) FROM (DUMMY);           00001010
                                 PUT DATA (DUM(I,2));                      00001020
                                 DUM(I,2) = 'REPLACED';                    00001030
                                 GO TO NXT;                                00001040
                                 END;                                      00001050
        END LOOP4;                                                         00001060
LBL1:   IF K = 0 THEN GO TO LBL2;                                          00001070
LOOP4A: DO I = 1 TO K;                                                     00001080
        IF DUMMY = REC(I) THEN DO;                                         00001090
                               REC(I) = 'DELETED';                         00001100
                               PUT LIST ('DELETED');                       00001110
                               GO TO NXT;                                  00001120
                               END;                                        00001130
        END LOOP4A;                                                        00001140
LBL2:   WRITE FILE (TEMP) FROM (DUMMY);                                    00001150
        GO TO NXT;                                                         00001160
EL2:                                                                       00001170
    ;                                                                      00001180
 STOP:  END EDIT;                                                          00001190
```

DOCUMENT NUMBER EDIT

```
VTISSUE: PROCEDURE OPTIONS (MAIN);                                    001010
      DCL       1 BUFFER_DISC,                                        001020
                2 DISC_KEY CHAR (6),                                  001030
                2 DISC_INFO  CHAR (12),                               001040
                1 CARD_IN,                                            001050
                2 CARD_KEY CHAR (6),                                  001070
                2 ACTUAL_INFO CHAR (12),                              001080
                2 FILL CHAR (61),                                     001090
                2 CHECK_1  CHAR (1);                                  001100
      DCL       END_CDE FIXED BINARY (15,0), CARD_HOLD CHAR (18);     001110
                END_CDE=0;                                            001120
                ON ENDFILE (DISC) GO TO EOJ2;                         001130
                READ FILE (DISC) INTO (BUFFER_DISC);                  001140
      READ1:    CARD_HOLD=' ';                                        001150
                ON ENDFILE (CARDS) GO TO EOJ1;                        001160
                READ FILE (CARDS) INTO (CARD_IN);                     001170
                CARD_HOLD=CARD_KEY||ACTUAL_INFO;                      001180
                IF CHECK_1= 'A' THEN GO TO ADDT;                      001190
      DUPS:     IF CARD_KEY = DISC_KEY THEN GO TO READ_DISC;          001200
                IF CARD_KEY > DISC_KEY THEN                           001210
                WRITE FILE (TAPE) FROM (BUFFER_DISC);                 001220
                IF CARD_KEY < DISC_KEY THEN GO TO READ1;              001230
      READ_DISC: READ FILE (DISC) INTO (BUFFER_DISC);                 001240
                GO TO DUPS;                                           001250
      ADDT:     IF CARD_KEY > DISC_KEY THEN DO;                       002010
                WRITE FILE (TAPE) FROM (CARD_HOLD);                   002020
                GO TO READ1;                                          002030
                  END;                                                002031
                ELSE WRITE FILE (TAPE) FROM (BUFFER_DISC);            002040
                READ FILE (DISC) INTO (BUFFER_DISC);                  002050
                GO TO ADDT;                                           002060
      EOJ1:     END_CDE=1;                                            002070
                READ FILE (DISC) INTO (BUFFER_DISC);                  002080
                WRITE FILE (TAPE)FROM (BUFFER_DISC);                  002090
                GO TO EOJ1;                                           002100
      EOJ2:     IF END_CDE = 1 THEN GO TO CLOSE_FILES;                002110
                ON ENDFILE (CARDS) GO TO CLOSE_FILES;                 002120
                CARD_HOLD=' ';                                        002130
                READ FILE (CARDS) INTO (CARD_IN);                     002140
                CARD_HOLD=CARD_KEY||ACTUAL_INFO;                      002150
                WRITE FILE (TAPE) FROM (CARD_HOLD);                   002160
                GO TO EOJ2;                                           002170
      CLOSE_FILES: CLOSE FILE (CARDS), FILE (DISC), FILE (TAPE);      002180
                END;                                                  002190
```

RIE FORMAT

```
/*

        /*
        PURPOSE: REFORMATING OF BIBLIOGRAPHIC AND ABSTRACT TAPES INTO
                 RIE FORMAT, THUS ALLOWING QUERY TO SEARCH THE DATA BASE.
        */
  CRIE: PROCEDURE OPTIONS(MAIN);                                          00000010
  DCL                                                                     00000020
       CODE_TABLE(20) FIXED BINARY(15,0), /* WILL BE TREATED AS BIT */    00000030
       BUFFER CHAR(6996) VAR,             /* USED AS I/O WORK AREA  */    00000040
      (LENGTHS,LEN_P_AUTHOR(2),#_P_AUTHORS,TITLE_LEN,#_C_AUTHORS,         00000050
       LEN_C_AUTHOR(5),#SA,LEN_SA(3),LEN_PRICE,LEN_CONTRACT,LEN_GRANT,    00000060
       LEN_PA,LEN_REPORT,LEN_AVAIL,LEN_JC,LEN_DATE,LEN_PAGES,LEN_DESC,    00000070
       LEN_DESC_NOTE,LEN_IDENT) FIXED BINARY(15,0),                       00000080
      VT CHAR(10) VAR,                                                    00000090
      ED CHAR(20) VAR,                                                    00000100
       PERSONAL_AUTHOR CHAR(264) VAR,                                     00000110
       TITLE           CHAR(396) VAR,                                     00000120
       CORPORATE_AUTHOR CHAR(660) VAR,                                    00000130
       SPONSOR_AGENCY  CHAR(396) VAR,                                     00000140
       PA              CHAR(2),                                           00000150
      (PRICE,CONTRACT,GRANT,REPORT,BUREAU) CHAR(66) VAR,                  00000160
       DATE            CHAR(7),                                           00000170
       PAGES           CHAR(16)  VAR,                                     00000180
      (DESC_NOTE,JOURNAL) CHAR(132) VAR,                                  00000190
       AVAIL           CHAR(264) VAR,                                     00000200
      DESCRIPTOR       CHAR(2000) VAR,                                    00000210
      EOF FIXED BINARY(15,0),                                             00000220
      ISSUE CHAR(15) VAR,                                                 00000230
      ISSUEQ CHAR(2),                                                     00000240
       IDENTIFIER      CHAR(396) VAR,                                     00000250
       BLANKS          CHAR(660) INITIAL((660)' '),                       00000260
      CODE_CHAR(20) CHAR(2) DEFINED CODE_TABLE,                           00000270
      PA_LEN CHAR(2) DEFINED LEN_PA,                                      00000280
      LENGTHS_CHAR CHAR(2) DEFINED LENGTHS,                               00000290
      DATE_LEN CHAR(2) DEFINED LEN_DATE,                                  00000300
      I_CHAR CHAR(2) DEFINED I,                                           00000310
      LEN_CHAR CHAR(2) DEFINED LEN_DESC,                                  00000320
       SHORT_TITLE     CHAR(396) VAR;                                     00000330
                                                                          00000340
   /*                                                                     00000350
        E O F CONTROL                                                     00000360
   */                                                                     00000370
                                                                          00000380
  ON ENDFILE(RIE) GO TO STOP;                                             00000390
                                                                          00000400
   /*                                                                     00000410
        SET UP C O D E  T A B L E                                         00000420
   */                                                                     00000430
                                                                          00000440
  CODE_TABLE(1)=16; CODE_TABLE(2)=17; CODE_TABLE(4)=23; CODE_TABLE(5)=26;00000450
  CODE_TABLE(6)=27; CODE_TABLE(19)=128; CODE_TABLE(20)=132;               00000460
  DO I=7 TO 10; CODE_TABLE(I)=28+I; END;                                  00000470
  DO I=11 TO 18;  CODE_TABLE(I)=32+I; END; CODE_TABLE(3)=20;              00000480
        OPEN FILE(ABSTRACT) SEQUENTIAL BUFFERED RECORD  INPUT;            00000490
                                                                          00000500
   /*                                                                     00000510
        READ FILE  R I E  INTO BUFFER -- WORK AREA                        00000520
   */                                                                     00000530
                                                                          00000540
```

```
/*
                                                                        00000550
   INPUT:                                                               00000560
  READ FILE(RIE) INTO(BUFFER); I=INDEX(BUFFER,'||')+1;                  00000570
 LEN_P_AUTHOR=0; LEN_C_AUTHOR=0; LEN_SA=0;                              00000580
                                                                        00000590
   /*                                                                   00000600
           DETERMINE WHETHER RECORD IS AIM OR ARM ISSUE                 00000610
   */                                                                   00000620
                                                                        00000630
     IF SUBSTR(BUFFER,48,1)='A' THEN ISSUE='AIM ';                      00000640
                                ELSE ISSUE='ARM ';                      00000650
                                                                        00000660
   /*                                                                   00000670
           DETERMINE QUARTER OF ISSUE                                   00000680
   */                                                                   00000690
                                                                        00000700
     ISSUEQ=SUBSTR(BUFFER,49,2);                                        00000710
 IF ISSUEQ='F ' THEN ISSUE=ISSUE||'FALL'||SUBSTR(BUFFER,51,4);          00000720
                ELSE                                                    00000730
  IF ISSUEQ='W ' THEN ISSUE=ISSUE||'WINTER'||SUBSTR(BUFFER,51,4);       00000740
                ELSE                                                    00000750
  IF ISSUEQ='SP' THEN ISSUE=ISSUE||'SPRING'||SUBSTR(BUFFER,51,4);       00000760
                ELSE                                                    00000770
 IF ISSUEQ='SU' THEN ISSUE=ISSUE||'SUMMER'||SUBSTR(BUFFER,51,4);        00000780
                                                                        00000790
   /*                                                                   00000800
           GET STRING IN BUFFER AND EDIT INFO ON IT                     00000810
   */                                                                   00000820
 GET STRING(BUFFER) EDIT(#_P_AUTHORS,(LEN_P_AUTHOR(J) DO J=1 TO #_P_AU00000830
  THORS),TITLE_LEN,#_C_AUTHORS,(LEN_C_AUTHOR(J) DO J=1 TO #_C_AUTHORS), 00000840
  #SA,(LEN_SA(J) DO J=1 TO #SA),LEN_PRICE,LEN_CONTRACT,LEN_GRANT,LEN_PA, 00000850
  LEN_REPORT,LEN_BUREAU,LEN_AVAIL,LEN_JC,LEN_DATE,LEN_PAGES,LEN_DESC_NOTE00000860
  ,LEN_DESC,LEN_IDENT)(X(I),28 F(4));                                   00000870
                                                                        00000880
   /*                                                                   00000890
           GET VT AND ED NUMBERS                                        00000900
   */                                                                   00000910
                                                                        00000920
  GET STRING(BUFFER) EDIT(VT,ED)(A(10),X(17),A(20));                    00000930
      J=58;                                                             00000940
      K=SUM(LEN_P_AUTHOR);                                              00000950
                                                                        00000960
   /*                                                                   00000970
           CHECK IF THERE ARE PERSONAL AUTHORS                          00000980
                IF SO, GET P. A. AND TITLE                              00000990
                IF NOT, GET TITLE                                       00001000
   */                                                                   00001010
                                                                        00001020
     IF K>0 THEN                                                        00001030
  GET STRING(BUFFER) EDIT(PERSONAL_AUTHOR,TITLE)(X(J),A(K),A(TITLE_LEN));00001040
                ELSE                                                    00001050
  GET STRING(BUFFER) EDIT(TITLE)(X(J),A(TITLE_LEN));                    00001060
      J=J+K+TITLE_LEN;                                                  00001070
      K=SUM(LEN_C_AUTHOR);                                              00001080
                                                                        00001090
   /*                                                                   00001100
```

```
/*
                                                                        00001110
          CHECK IF THERE ARE CORPORATE AUTHORS                          00001120
                IF SO, GET C.A.                                         00001130
   */                                                                   00001140
                                                                        00001150
      IF K>0 THEN DO;                                                   00001160
  GET STRING(BUFFER) EDIT(CORPORATE_AUTHOR)(X(J),A(K));                 00001170
      J=J+K;                                                            00001180
                END;                                                    00001190
      K=SUM(LEN_SA);                                                    00001200
                                                                        00001210
   /*                                                                   00001220
          CHECK IF THERE ARE SPONSORING AGENCIES                        00001230
                IF SO, GET S.A.                                         00001240
   */                                                                   00001250
      IF K>0 THEN DO;                                                   00001260
  GET STRING(BUFFER) EDIT(SPONSOR_AGENCY)(X(J),A(K));                   00001270
      J=J+K;                                                            00001280
                END;                                                    00001290
                                                                        00001300
   /*                                                                   00001310
          GET STRING IN BUFFER AND EDIT MORE INFORMATION ON FILE        00001320
   */                                                                   00001330
                                                                        00001340
  GET STRING(BUFFER) EDIT(                                              00001350
                PRICE,CONTRACT,GRANT,PA,REPORT,BUREAU,AVAIL,JOURNAL,DATE00001360
  ,PAGES,DESC_NOTE,DESCRIPTOR,IDENTIFIER)(X(J),A(LEN_PRICE),            00001370
       A(LEN_CONTRACT),A(LEN_GRANT),A(LEN_PA),A(LEN_REPORT),A(LEN_BUREAU00001380
  ),A(LEN_AVAIL),A(LEN_JO),A(LEN_DATE),A(LEN_PAGES),A(LEN_DESC_NOTE),   00001390
  A(LEN_DESC),A(LEN_IDENT));                                            00001400
                                                                        00001410
   /*                                                                   00001420
          CHECK IF THERE IS AN ASSIGNED ED NUMBER                       00001430
                IF SO, GO TO R I E INPUT                                00001440
                IF NOT, CONTINUE                                        00001450
   */                                                                   00001460
                                                                        00001470
     IF SUBSTR(ED,3,1)='('|SUBSTR(ED,11,1)¬=' ' THEN DO;                00001480
               GO TO INPUT;                                             00001490
                                                    END;                00001500
  LENGTHS=12;                                                           00001510
   ED=SUBSTR(ED,1,2)||SUBSTR(ED,4,3)||SUBSTR(ED,8,3);                   00001520
  VT=SUBSTR(VT,1,2)||SUBSTR(VT,4,3)||SUBSTR(VT,8,3);                    00001530
                                                                        00001540
   /*                                                                   00001550
          CHECK IF THERE IS AN ED NUMBER                                00001560
                IF SO, PLACE VT AND ED CODES, LENGTHS, AND NUMBERS      00001570
                INTO BUFFER                                             00001580
                IF NOT, PLACE VT C, L, AND N INTO BUFFER                00001590
   */                                                                   00001600
                                                                        00001610
   IF SUBSTR(ED,4,1)¬=' ' THEN DO;                                      00001620
     BUFFER=LENGTHS_CHAR||CODE_CHAR(1)||ED;                             00001630
     BUFFER=BUFFER||LENGTHS_CHAR||CODE_CHAR(2)||VT;                     00001640
     END;                                                               00001650
    ELSE                                                                00001660
```

```
/*
```

```
   BUFFER=LENGTHS_CHAR||CODE_CHAR(2)||VT;                                    00001670
                                                                             00001680
   /*                                                                        00001690
            CHECK IF THERE IS A PROGRAM AREA                                 00001700
                  IF SO, PUT P.A. C, L, AND N INTO BUFFER                    00001710
   */                                                                        00001720
                                                                             00001730
   IF LEN_PA¬=0 THEN DO;                                                     00001740
      LEN_PA=LEN_PA+4;                                                       00001750
      BUFFER=BUFFER||PA_LEN||CODE_CHAR(3)||PA;                               00001760
   END;                                                                      00001770
                                                                             00001780
   /*                                                                        00001790
            CHECK IF THERE IS A DATE                                         00001800
                  IF SO, PUT DATE CODE, LENGTH, AND DATA INTO BUFFER         00001810
   */                                                                        00001820
                                                                             00001830
   IF DATE¬='        '&DATE¬='        ' THEN DO;                             00001840
      LEN_DATE=LEN_DATE+4;                                                   00001850
      UNSPEC(DATE)=UNSPEC(DATE)||UNSPEC(SUBSTR(BLANKS,1,LEN_DATE-4));       00001860
      BUFFER=BUFFER||DATE_LEN||CODE_CHAR(4)||DATE;                           00001870
                                                 END;                        00001880
                                                                             00001890
   /*                                                                        00001900
            CHECK IF THERE IS A TITLE                                        00001910
                  IF SO, PUT TITLE C, L, AND DATA INTO BUFFER                00001920
   */                                                                        00001930
                                                                             00001940
   IF TITLE_LEN¬=0 THEN DO;                                                  00001950
      UNSPEC(TITLE)=UNSPEC(TITLE)||UNSPEC(SUBSTR(BLANKS,1,TITLE_LEN));      00001960
BTITLE: I=INDEX(TITLE,'<');                                                  00001970
        IF I=0 THEN GO TO ENDTITLE;                                          00001980
               ELSE IF I=1 THEN TITLE=SUBSTR(TITLE,2);                       00001990
                           ELSE TITLE=SUBSTR(TITLE,1,I-1)||SUBSTR(TITLE00002000
                                    ,I+1);                                   00002010
        I=INDEX(TITLE,'>');                                                  00002020
        IF I=0 THEN DO;                                                      00002030
                    PUT FILE(SYSPRINT) LIST('** TITLE ERROR**',VT);          00002040
                    GO TO INPUT;                                             00002050
                    END;                                                     00002060
               ELSE TITLE=SUBSTR(TITLE,1,I-1)||SUBSTR(TITLE,I+2);            00002070
        GO TO BTITLE;                                                        00002080
ENDTITLE: I=LENGTH(TITLE)+4;                                                 00002090
      BUFFER=BUFFER||I_CHAR||CODE_CHAR(5)||TITLE;                            00002100
                    END;                                                     00002110
                                                                             00002120
   /*                                                                        00002130
            CHECK IF THERE IS A PERSONAL AUTHOR                              00002140
                  IF SO, PUT P. A.  C, L, AND DATA INTO BUFFER               00002150
   */                                                                        00002160
                                                                             00002170
   IF #_P_AUTHORS>0 THEN DO;                                                 00002180
      I=SUM(LEN_P_AUTHOR);                                                   00002190
      UNSPEC(PERSONAL_AUTHOR)=UNSPEC(PERSONAL_AUTHOR)||UNSPEC(SUBSTR(        00002200
         BLANKS,1,I));                                                       00002210
      IF #_P_AUTHORS=2 THEN DO;                                              00002220
```

```
/*
                                                                        00002230
        SUBSTR(PERSONAL_AUTHOR, LEN_P_AUTHOR(1)+1)=';'||SUBSTR(         00002240
            PERSONAL_AUTHOR,LEN_P_AUTHOR(1)+1);                         00002250
        I=I+1;                                                          00002260
                              END;                                      00002270
    I=I+4;                                                              00002280
    BUFFER=BUFFER||I_CHAR||CODE_CHAR(6)||PERSONAL_AUTHOR;               00002290
                          END;                                          00002300
                                                                        00002310
  /*                                                                    00002320
        BLOCK FOR MANIPULATING DESCRIPTOR CODE, LENGTH, AND DATA        00002330
  */                                                                    00002340
                                                                        00002350
    LEN_DESC_NOTE=LEN_DESC_NOTE+LEN_PAGES;                              00002360
    DESC_NOTE=PAGES||DESC_NOTE;                                         00002370
DO I=7 TO 18;                                                           00002380
    IF I=8 THEN DO; LEN_DESC=LEN_IDENT;                                 00002390
                    DESCRIPTOR=IDENTIFIER;                              00002400
                END;                                                    00002410
          ELSE                                                          00002420
    IF I=9 THEN DO; LEN_DESC=LEN_PRICE;                                 00002430
                    DESCRIPTOR=PRICE;                                   00002440
                END;                                                    00002450
          ELSE                                                          00002460
    IF I=10 THEN DO; LEN_DESC=LEN_DESC_NOTE;                            00002470
                     DESCRIPTOR=DESC_NOTE;                              00002480
                END;                                                    00002490
          ELSE                                                          00002500
     IF I=11 THEN DO; LEN_DESC=LENGTH(ISSUE);                           00002510
                      DESCRIPTOR=ISSUE;                                 00002520
                END;                                                    00002530
      ELSE                                                              00002540
     IF I=12 THEN DO;                                                   00002550
       EOF=1;                                                           00002560
     ON ENDFILE(ABSTRACT) BEGIN;                                        00002570
       PUT FILE(SYSPRINT) EDIT(EOF,VT)(SKIP,F(4),A(10));                00002580
       IF EOF=2 THEN GO TO LOOPEND;                                     00002590
       EOF=2;                                                           00002600
       CLOSE FILE(ABSTRACT);                                            00002610
       OPEN FILE(ABSTRACT) SEQUENTIAL BUFFERED RECORD  INPUT;           00002620
       GO TO ABSL;                                                      00002630
                              END;                                      00002640
       GO TO ABVT;                                                      00002650
                END;                                                    00002660
     ELSE                                                               00002670
   IF I=13 THEN DO; LEN_DESC=LEN_REPORT;                                00002680
                    DESCRIPTOR=REPORT;                                  00002690
                END;                                                    00002700
          ELSE                                                          00002710
   IF I=14 THEN DO; LEN_DESC=LEN_CONTRACT;                              00002720
                    DESCRIPTOR=CONTRACT;                                00002730
                END;                                                    00002740
          ELSE                                                          00002750
   IF I=15 THEN DO; LEN_DESC=LEN_GRANT;                                 00002760
                    DESCRIPTOR=GRANT;                                   00002770
                END;                                                    00002780
          ELSE
```

```
/*
                                                                        00002790
    IF I=16 THEN DO; LEN_DESC=LEN_BUREAU;                               00002800
                     DESCRIPTOR=BUREAU;                                 00002810
                END;                                                    00002820
            ELSE                                                        00002830
    IF I=17 THEN DO; LEN_DESC=LEN_AVAIL;                                00002840
                     DESCRIPTOR=AVAIL;                                  00002850
                END;                                                    00002860
    IF I=18 THEN DO; LEN_DESC=LEN_JC;                                   00002870
                     DESCRIPTOR=JOURNAL;                                00002880
                END;                                                    00002890
        IF LEN_DESC¬=0 THEN DO;                                         00002900
          UNSPEC(DESCRIPTOR)=UNSPEC(DESCRIPTOR)||UNSPEC(SUBSTR(         00002910
              BLANKS,1,LEN_DESC));                                      00002920
  INSERT: LEN_DESC=LEN_DESC+4;                                          00002930
      BUFFER=BUFFER||LEN_CHAR||CODE_CHAR(I)||DESCRIPTOR;                00002940
              LEN_DESC=0;                                               00002950
      GO TO LOOPEND;                                                    00002960
                                                                        00002970
   /*                                                                   00002980
          END OF DESCRIPTOR BLOCK                                       00002990
   */                                                                   00003000
   /*                                                                   00003010
        CONSTRUCT ABSTRACT VT NUMBER                                    00003020
   */                                                                   00003030
  ABVT: VT='VT '||SUBSTR(VT,3,3)||' '||SUBSTR(VT,6,3);                  00003040
                                                                        00003050
   /*                                                                   00003060
        READ THE FILE ABSTRACT INTO DESCRIPTOR -- WORK AREA             00003070
   */                                                                   00003080
                                                                        00003090
  ABSL: READ FILE(ABSTRACT) INTO(DESCRIPTOR);                           00003100
   IF SUBSTR(DESCRIPTOR,1,10)¬=VT THEN GO TO ABSL;                      00003110
                                                                        00003120
   /*                                                                   00003130
          SPECIFY DESCRIPTOR DATA AND LENGTH OF DATA                    00003140
   */                                                                   00003150
                                                                        00003160
              DESCRIPTOR=SUBSTR(DESCRIPTOR,13);                         00003170
              LEN_DESC=LENGTH(DESCRIPTOR);                              00003180
      GO TO INSERT;                                                     00003190
  END;                                                                  00003200
LOOPEND: END;                                                           00003210
                                                                        00003220
   /*                                                                   00003230
          CHECK IF THERE ARE ANY CORPORATE AUTHORS                      00003240
        IF SO, PUT C.A. CODE, LENGTH, AND DATA INTO BUFFER              00003250
   */                                                                   00003260
                                                                        00003270
   IF #_C_AUTHORS>0 THEN DO;                                            00003280
      I=SUM(LEN_C_AUTHOR); J=0;                                         00003290
      UNSPEC(CORPORATE_AUTHOR)=UNSPEC(CORPORATE_AUTHOR)||UNSPEC(SUBSTR( 00003300
        BLANKS,1,I));                                                   00003310
      DESCRIPTOR=CORPORATE_AUTHOR;                                      00003320
      IF #_C_AUTHORS>1 THEN DO K=2 TO #_C_AUTHORS;                      00003330
         J=J+LEN_C_AUTHOR(K-1)+1;                                       00003340
```

```
/*
                                                                        00003350
          SUBSTR(DESCRIPTOR,J,LEN_C_AUTHOR(K)+1)=';'||SUBSTR           00003360
               CORPORATE_AUTHOR,LEN_C_AUTHOR(K-1),LEN_C_AUTHOR(K));     00003370
                         END;                                           00003380
       I=I+#_C_AUTHORS+3;                                               00003390
       BUFFER=BUFFER||I_CHAR||CODE_CHAR(19)||DESCRIPTOR;                00003400
                        END;                                            00003410
                                                                        00003420
    /*                                                                  00003430
          CHECK IF THERE ARE ANY SPONSORING AGENCIES                    00003440
               IF SO, PUT S.A. CODE, LENGTH, AND DATA INTO BUFFER       00003450
    */                                                                  00003460
                                                                        00003470
  IF #SA>0 THEN DO;                                                     00003480
       DESCRIPTOR=SPONSOR_AGENCY;                                       00003490
       I=SUM(LEN_SA);                                                   00003500
       UNSPEC(SPONSOR_AGENCY)=UNSPEC(SPONSOR_AGENCY)||UNSPEC(SUBSTR(BLANKS,00003510
         1,I));  J=0;                                                   00003520
      IF #SA>1 THEN DO K=2 TO #SA;                                      00003530
            J=J+LEN_SA(K-1)+1;                                          00003540
            SUBSTR(DESCRIPTOR,J,LEN_SA(K)+1)=';'||SUBSTR(               00003550
               SPONSOR_AGENCY,J-K+1,LEN_SA(K));                         00003560
                 END;                                                   00003570
       I=I+#SA+3;                                                       00003580
       BUFFER=BUFFER||I_CHAR||CODE_CHAR(20)||DESCRIPTOR;                00003590
               END;                                                     00003600
                                                                        00003610
    /*                                                                  00003620
          WRITE FILE BUFFER ONTO THE MASTER FILE                        00003630
    */                                                                  00003640
                                                                        00003650
  WRITE FILE(MASTER) FROM(BUFFER);                                      00003660
  GO TO INPUT;                                                          00003670
  STOP: END CRIE;
```

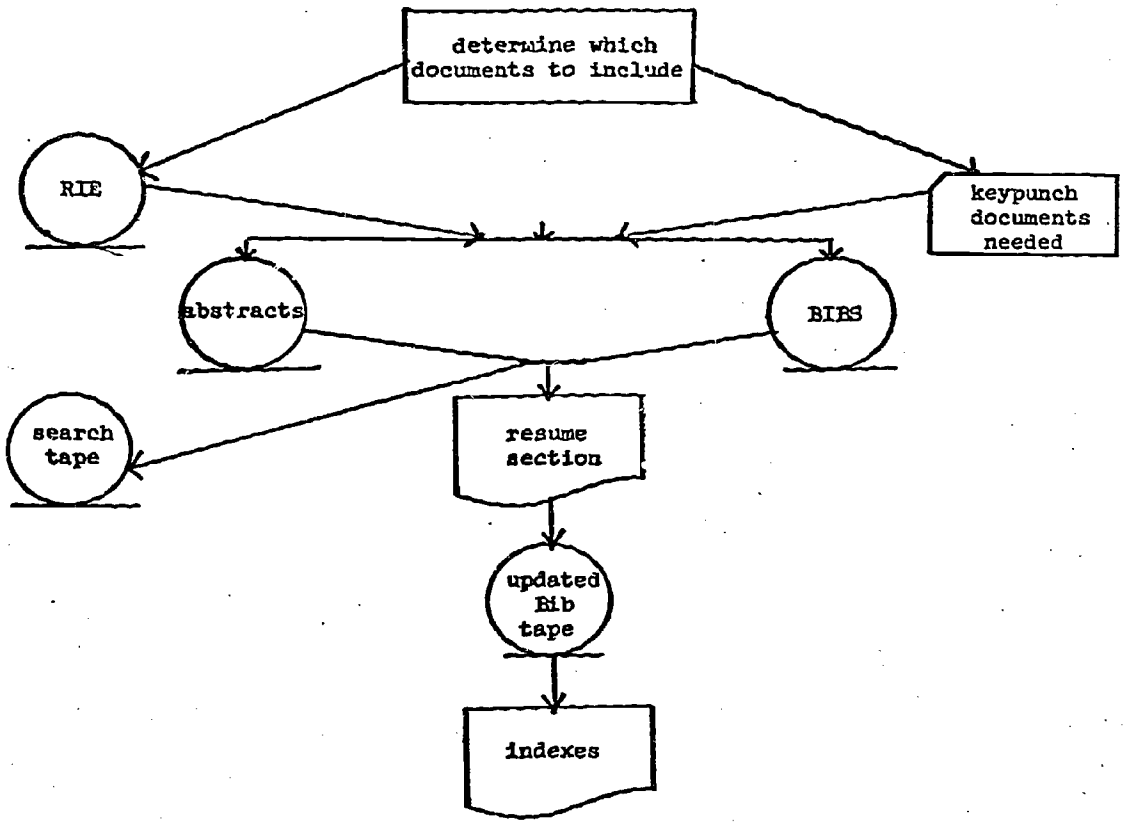
APPENDIX F

PROGRAM AND SYSTEM FLOWCHARTS

	page
OVERVIEW OF SYSTEM	211
DETAILED FLOWCHART OF SYSTEM	213
RIE TO AIM/ARM	218
MASTER	221
ABSTRACT	231
RESUME PRINT	234
MASTER EDIT	236
SUBJECT QUARTERLY	238
AUTHOR QUARTERLY	240
SUPPORTING SERVICES QUARTERLY	242
DOCUMENT NUMBER QUARTERLY	244
CONVERSION OF DOCUMENT NUMBER	246
SUBJECT ANNUAL	248
AUTHOR ANNUAL	250
SUPPORTING SERVICES ANNUAL	252
DOCUMENT NUMBER ANNUAL	254
EDIT OF SUBJECT, AUTHOR, AND SUPPORTING SERVICES	256
DOCUMENT NUMBER EDIT	258
RIE FORMAT	261

OVERVIEW OF SYSTEM

OVERVIEW OF THE SYSTEM



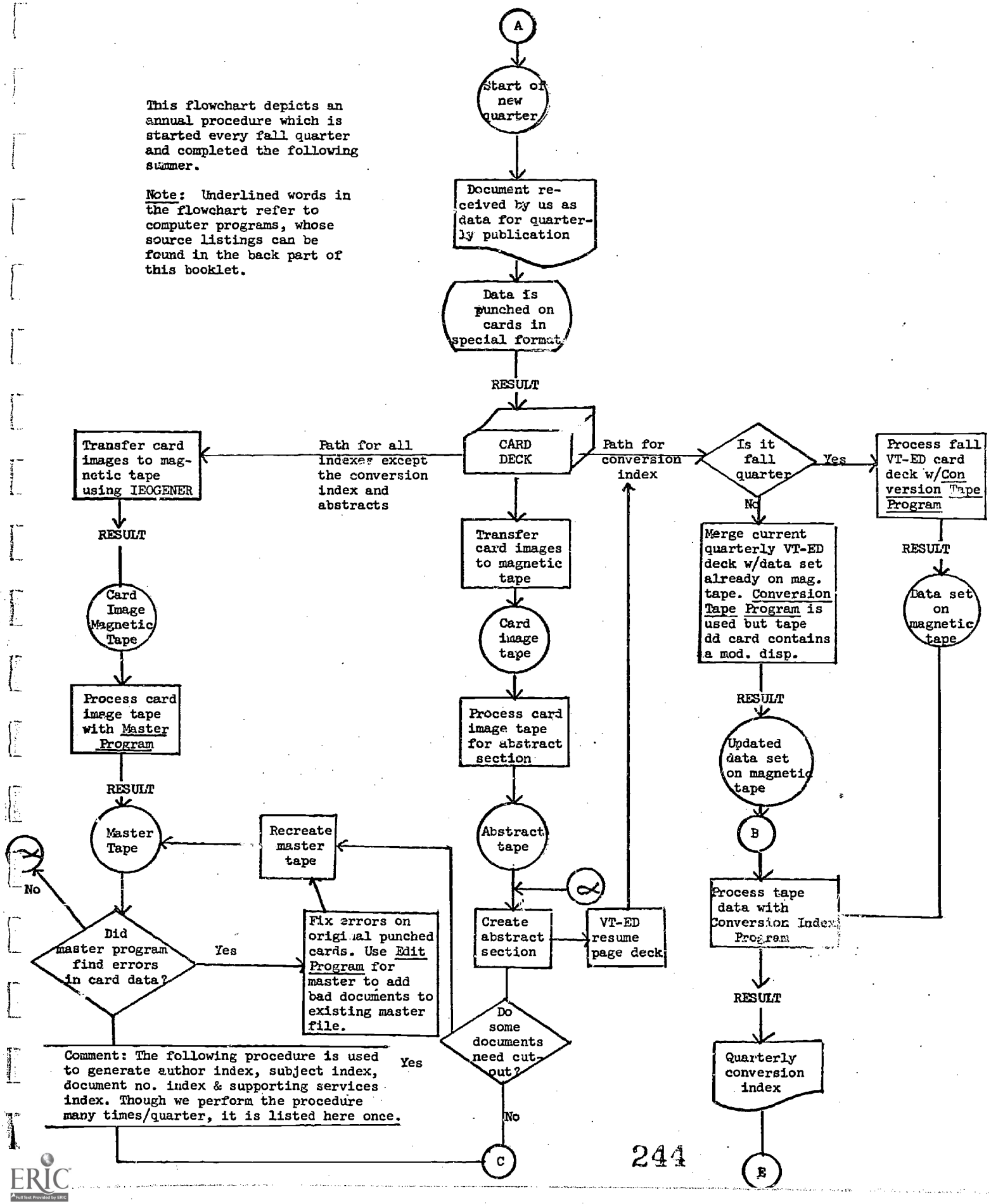
DETAILED FLOWCHART OF SYSTEM

Flowchart of the System

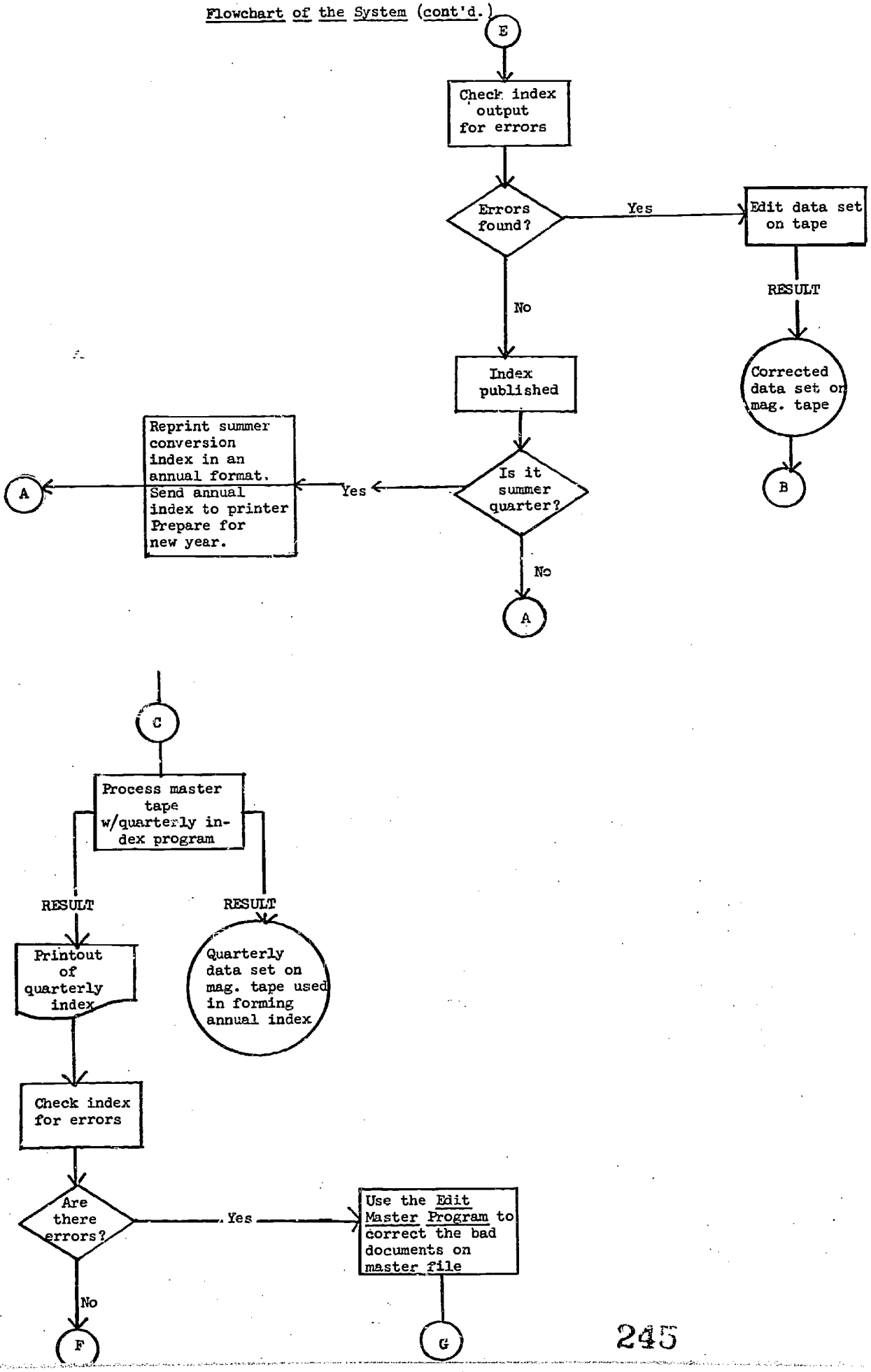
(The System Is the Same for AIM and ARM)

This flowchart depicts an annual procedure which is started every fall quarter and completed the following summer.

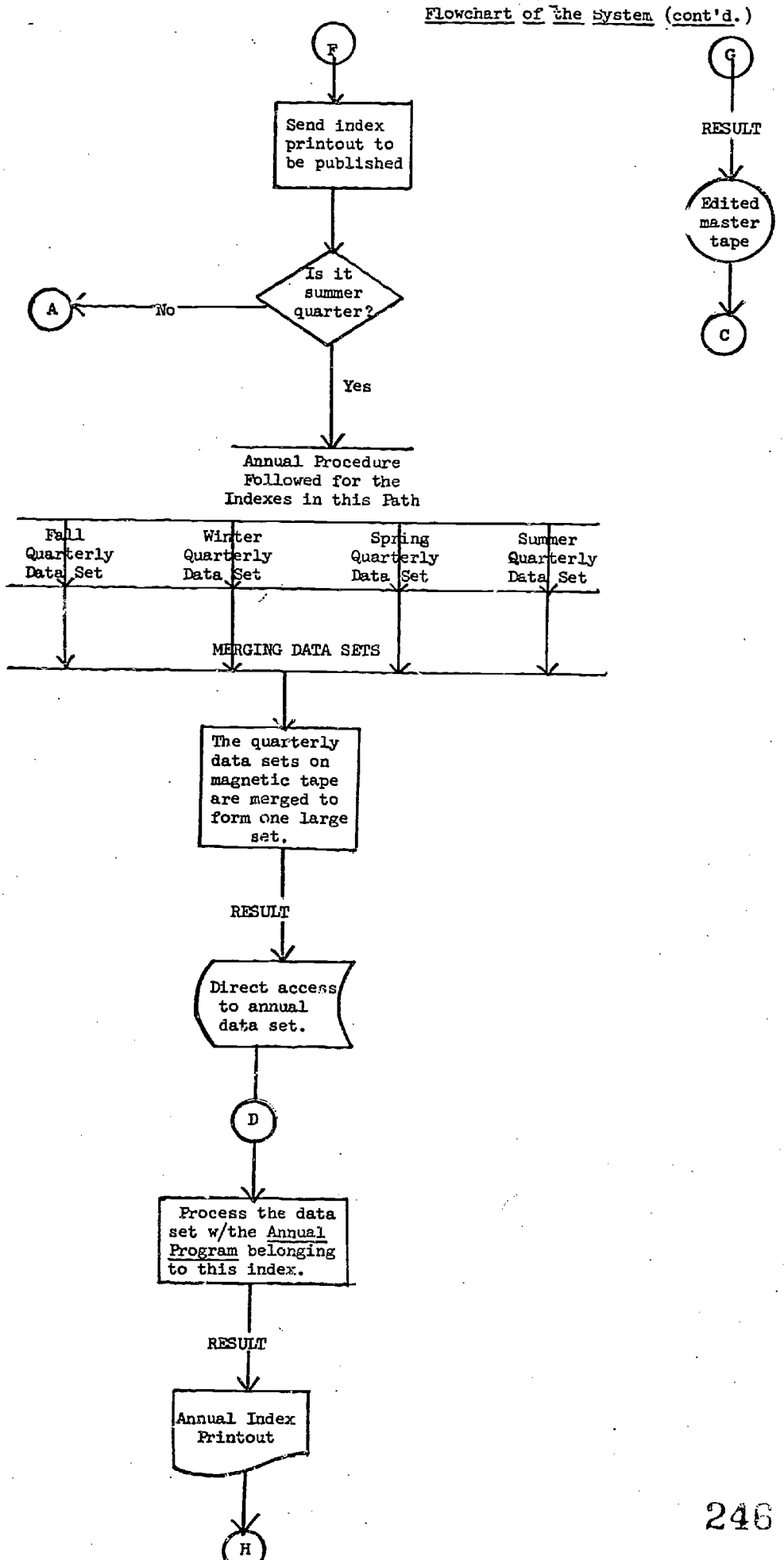
Note: Underlined words in the flowchart refer to computer programs, whose source listings can be found in the back part of this booklet.



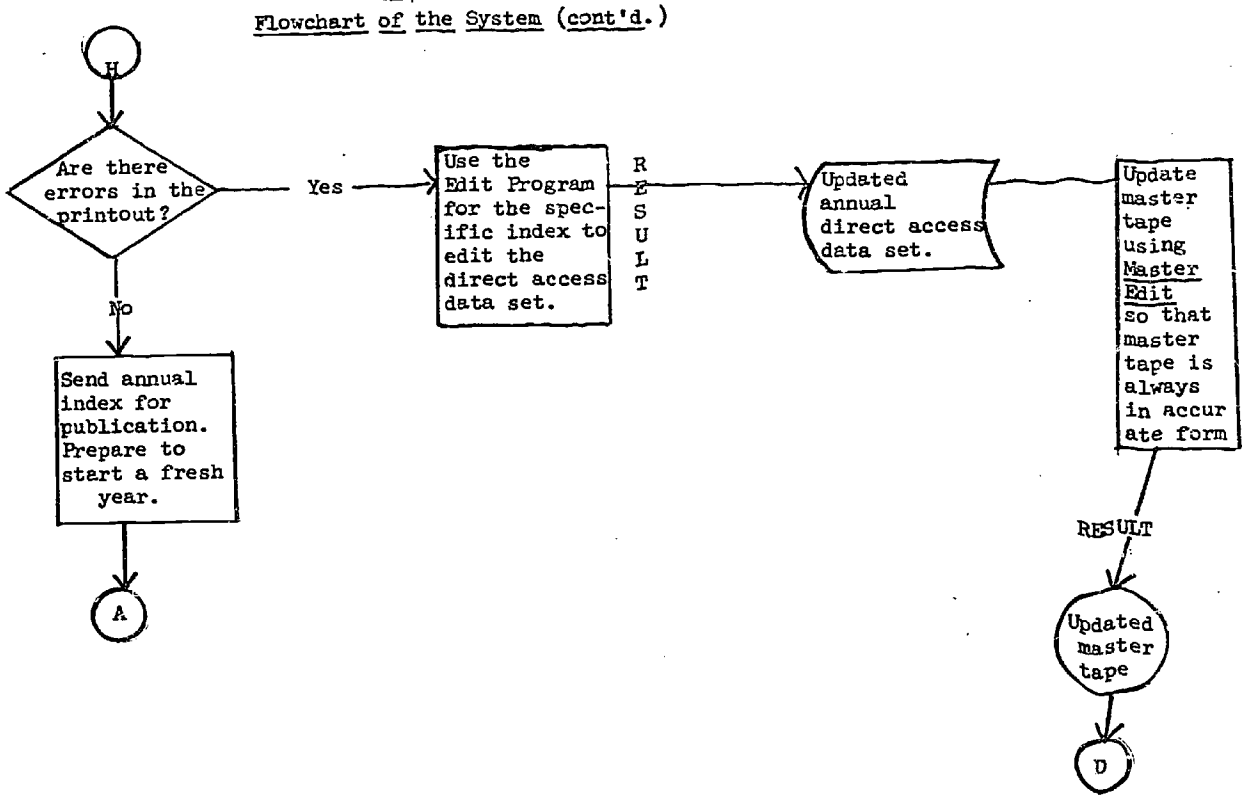
Flowchart of the System (cont'd.)



Flowchart of the System (cont'd.)

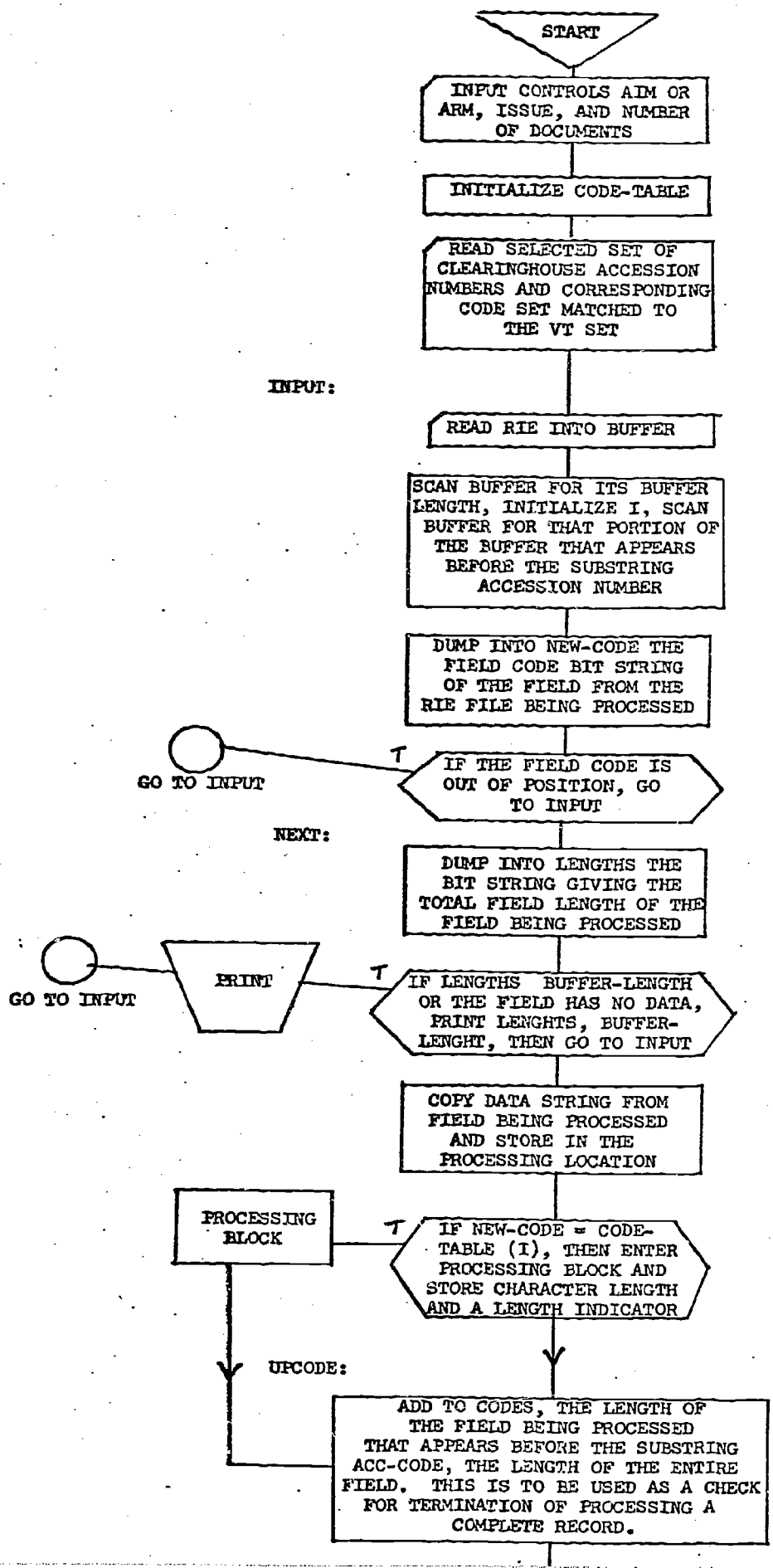


Flowchart of the System (cont'd.)

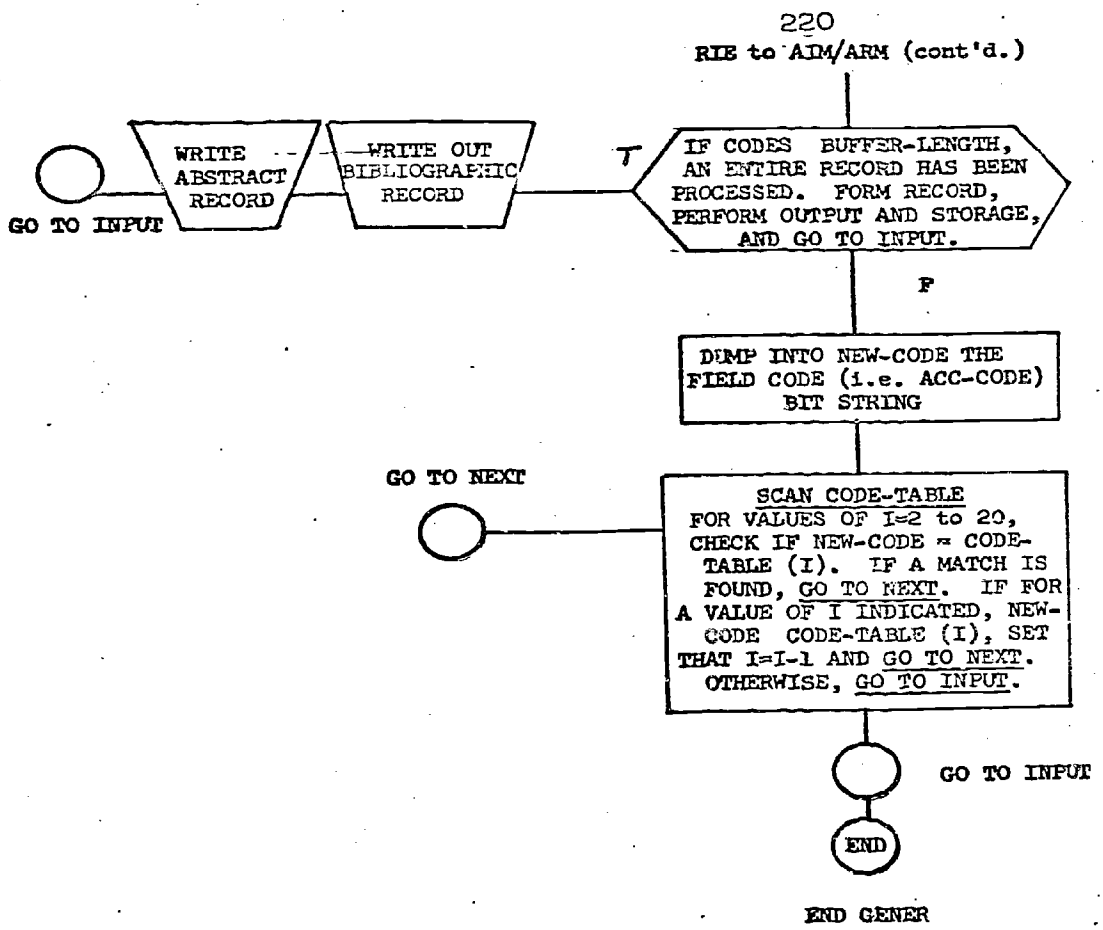


RIE TO AIM/ARM

RIE to AIM/ARM

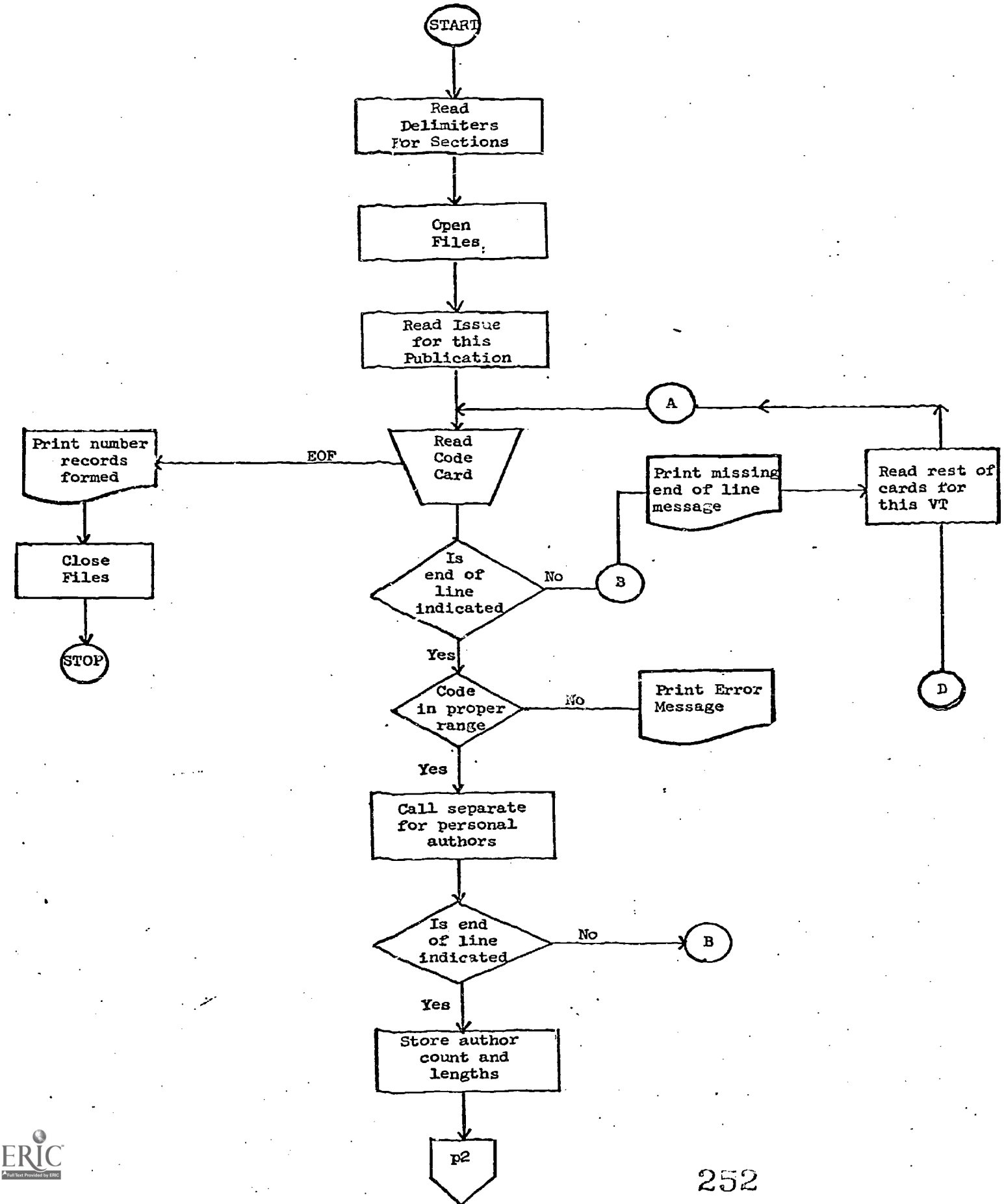


RIE to AIM/ARM (cont'd.)

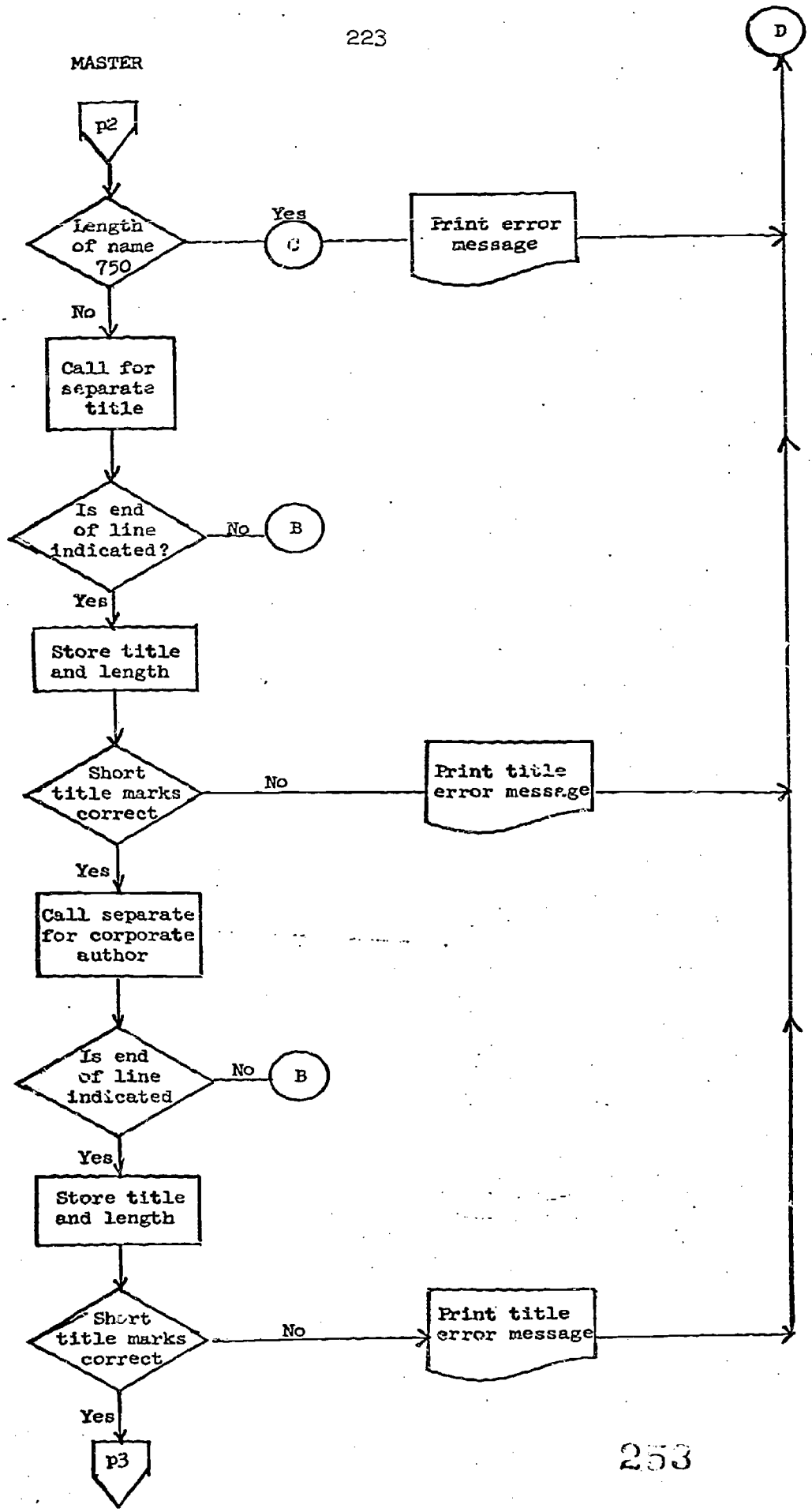


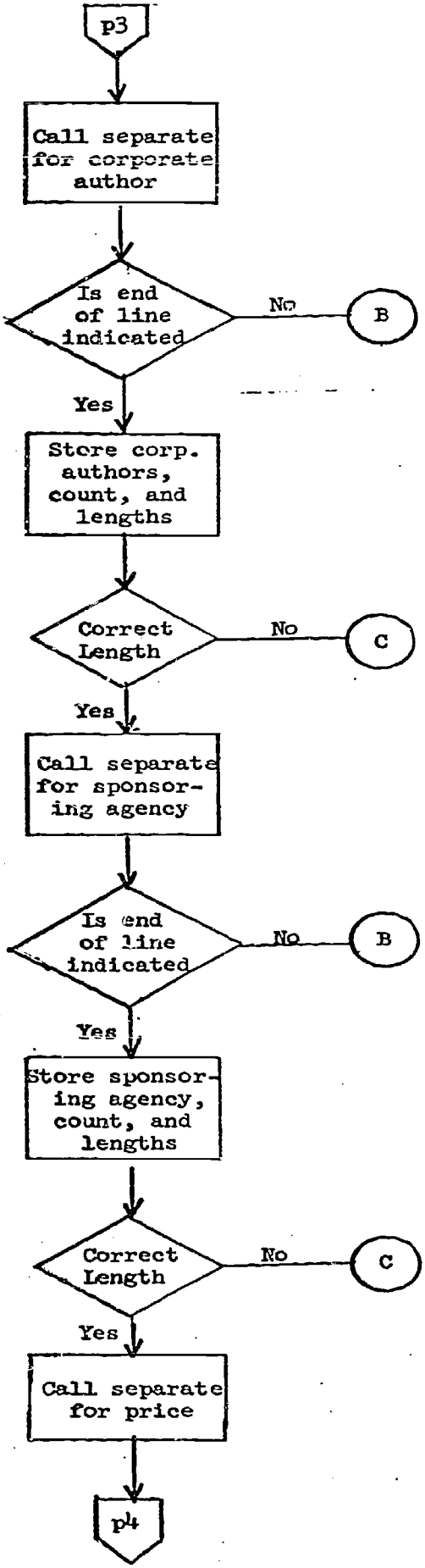
END GENER

MASTER

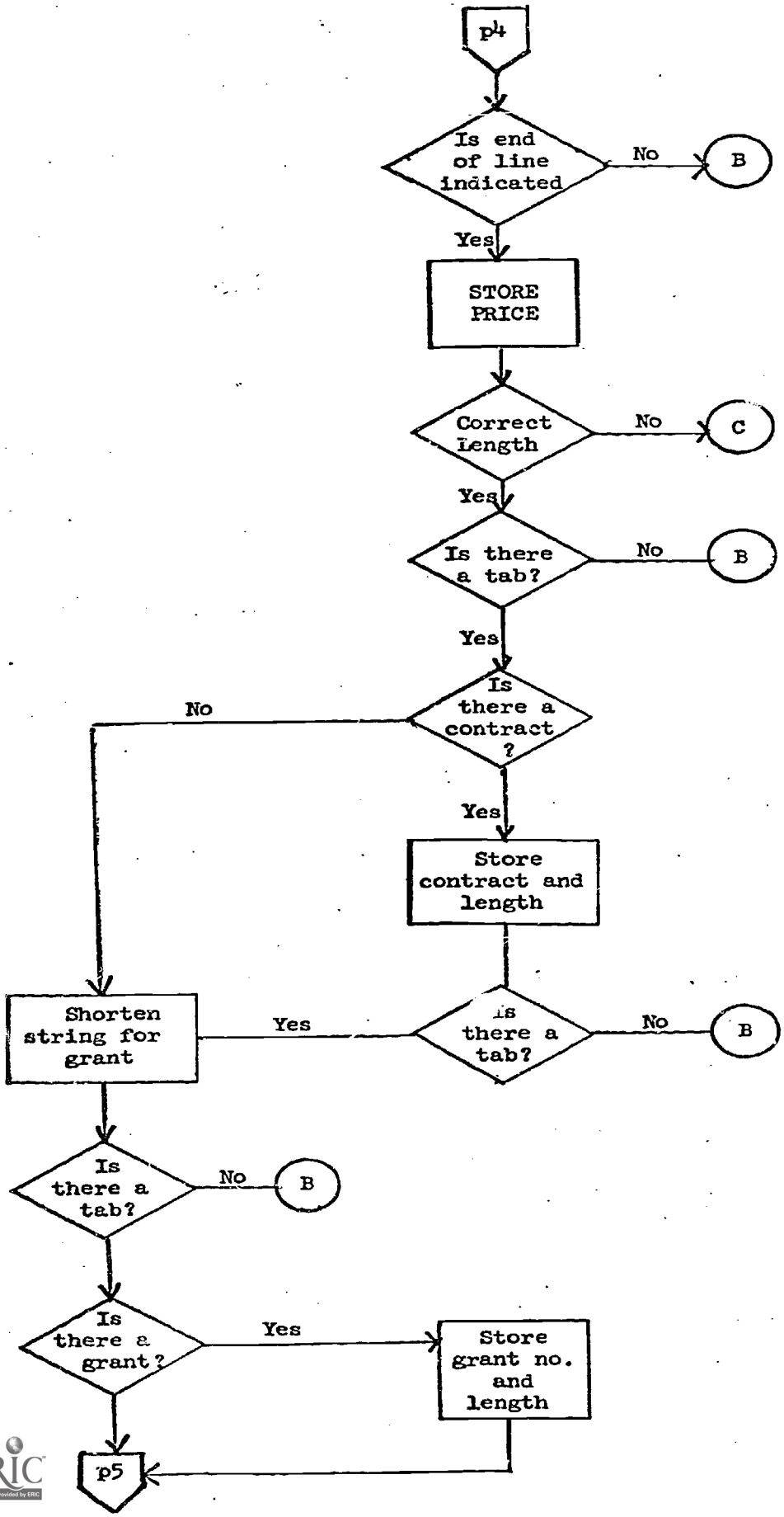

START
Read Delimiters For Sections
Open Files
Read Issue for this Publication
A
Read Code Card
EOF
Print number records formed
Close Files
STOP
Is end of line indicated
No
B
Print missing end of line message
Read rest of cards for this VT
D
Yes
Code in proper range
No
Print Error Message
Yes
Call separate for personal authors
Is end of line indicated
No
B
Yes
Store author count and lengths
p2

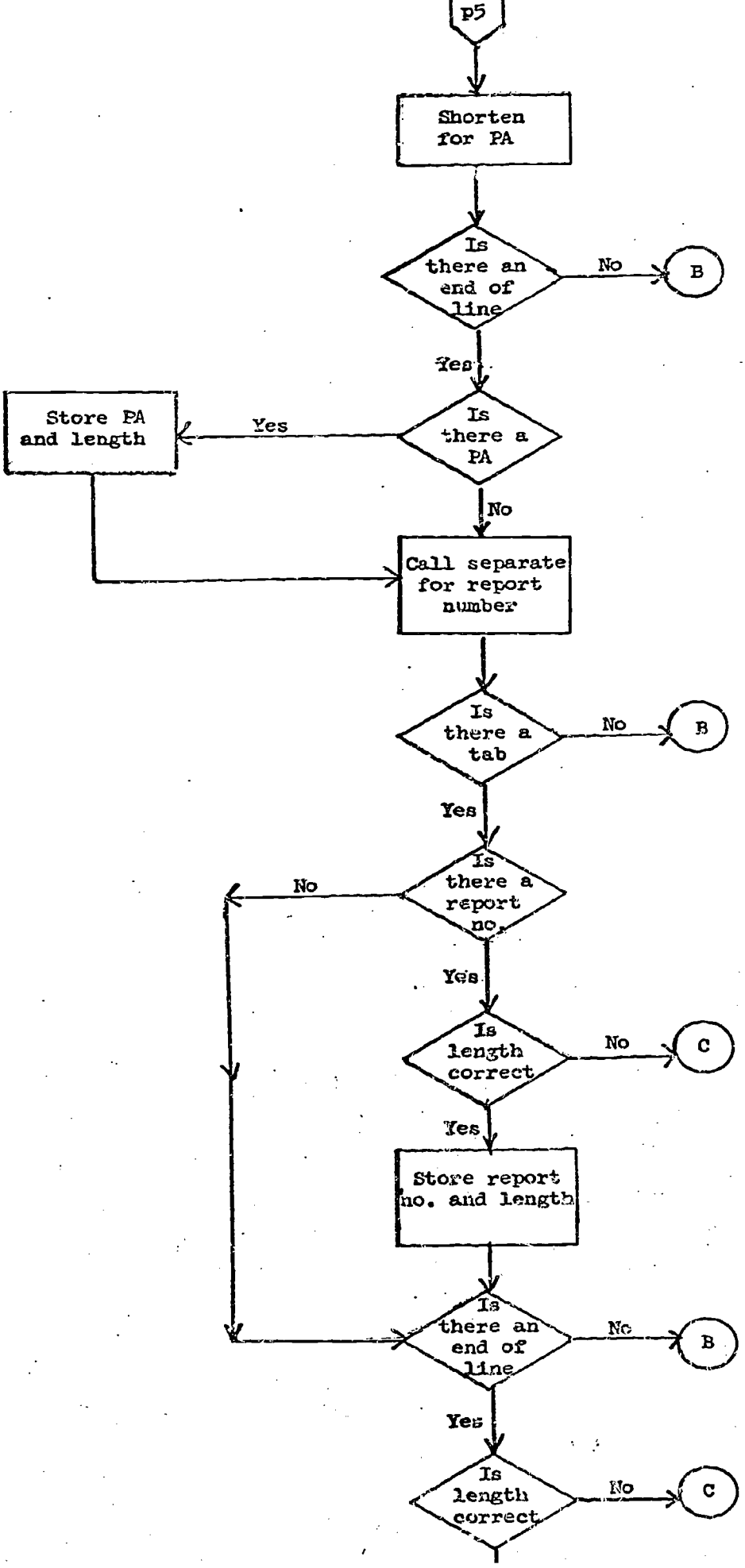
MASTER

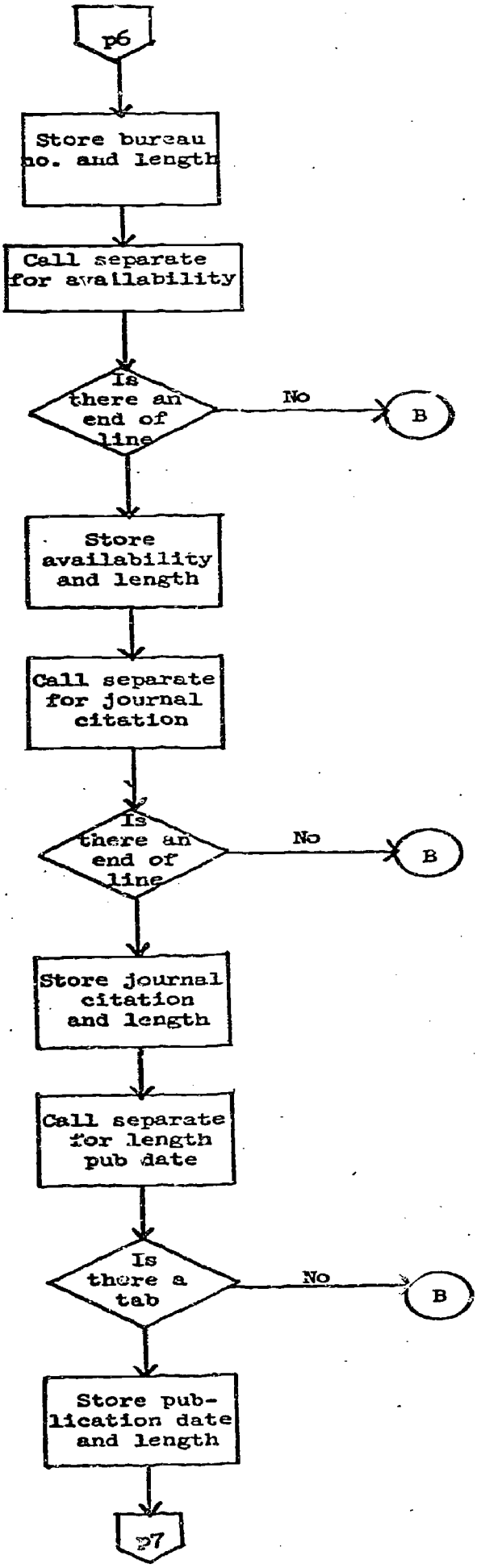

D
p2
Length of name 750
Yes
C
Print error message
No
Call for separate title
Is end of line indicated?
No
B
Yes
Store title and length
Short title marks correct
No
Print title error message
Yes
Call separate for corporate author
Is end of line indicated
No
B
Yes
Store title and length
Short title marks correct
No
Print title error message
Yes
p3

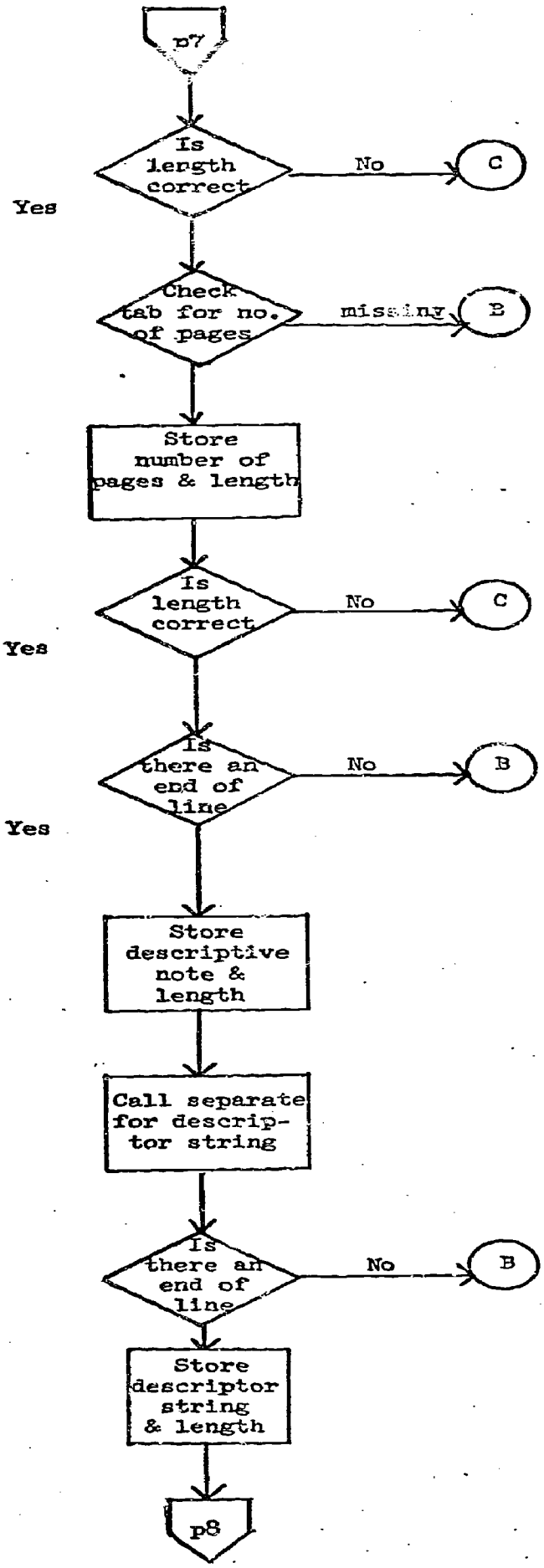

p3
Call separate for corporate author
Is end of line indicated
No
B
Yes
Store corp. authors, count, and lengths
Correct Length
No
C
Yes
Call separate for sponsoring agency
Is end of line indicated
No
B
Yes
Store sponsoring agency, count, and lengths
Correct Length
No
C
Yes
Call separate for price
p4

MASTER

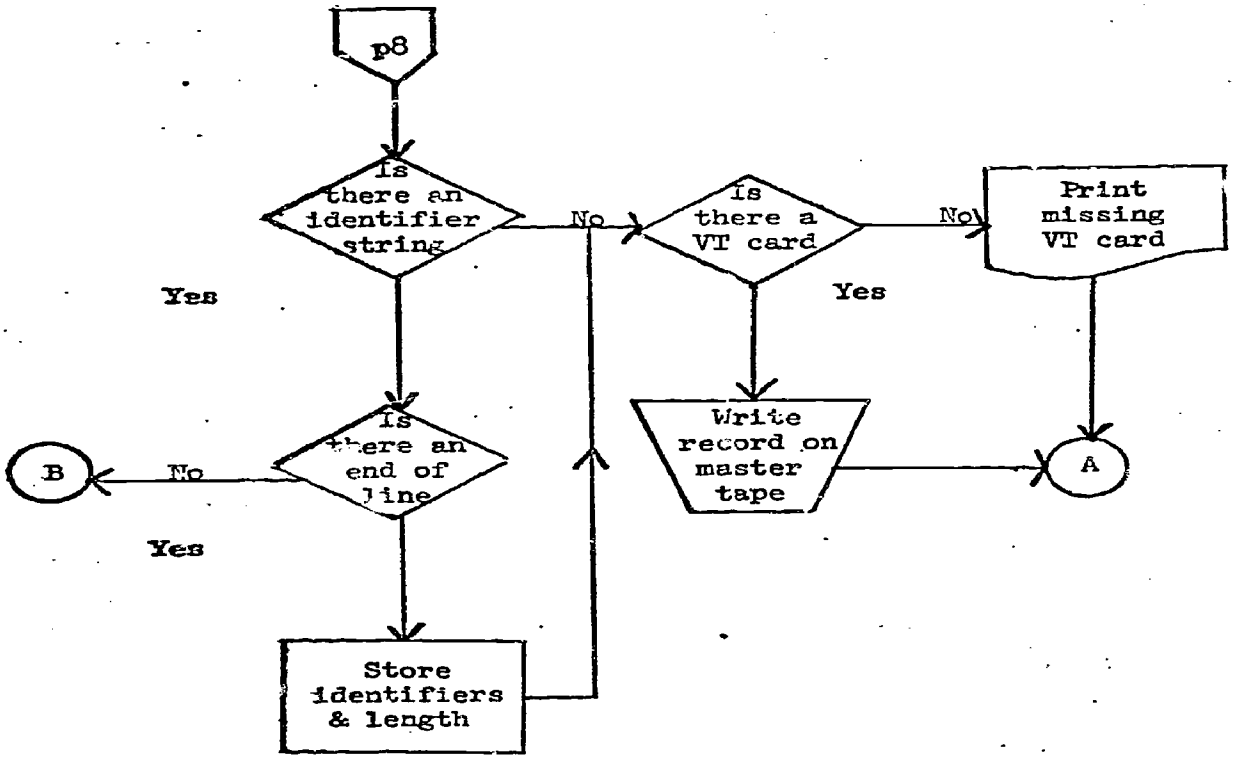

p4
Is end of line indicated
No
B
Yes
STORE PRICE
Correct length
No
C
Yes
Is there a tab?
No
B
Yes
Is there a contract ?
No
Yes
Store contract and length
Shorten string for grant
Yes
Is there a tab?
No
B
Is there a tab?
No
B
Is there a grant?
Yes
Store grant no. and length
p5

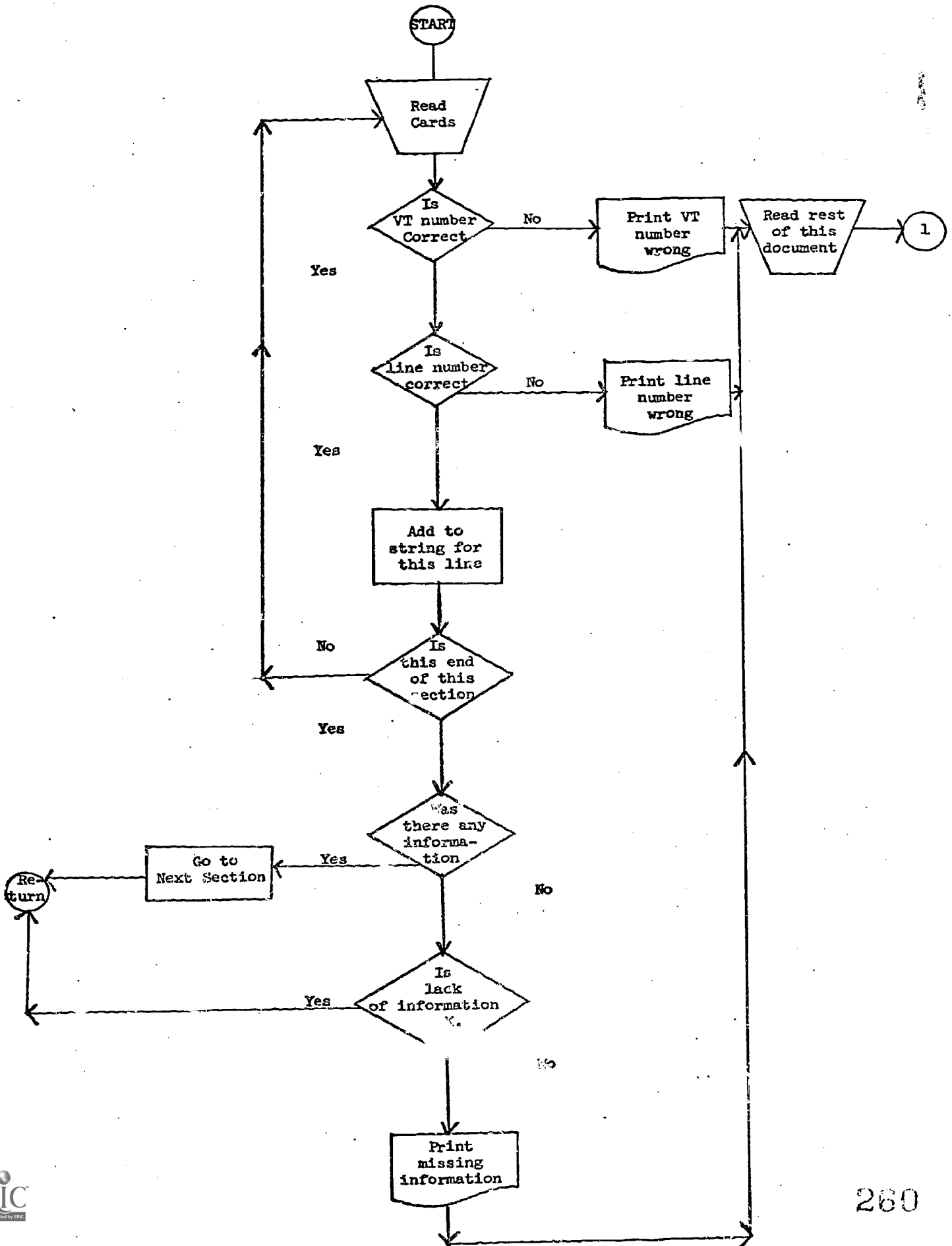

p5
Shorten for PA
Is there an end of line
No
B
Yes
Is there a PA
Yes
Store PA and length
No
Call separate for report number
Is there a tab
No
B
Yes
Is there a report no.
No
Yes
Is length correct
No
C
Yes
Store report no. and length
Is there an end of line
No
B
Yes
Is length correct
No
C


p6
Store bureau no. and length
Call separate for availability
Is there an end of line
No
B
Store availability and length
Call separate for journal citation
Is there an end of line
No
B
Store journal citation and length
Call separate for length pub date
Is there a tab
No
B
Store publication date and length
p7


p7
Is length correct
No
C
Yes
Check tab for no. of pages
missing
B
Store number of pages & length
Is length correct
No
C
Yes
Is there an end of line
No
B
Yes
Store descriptive note & length
Call separate for descriptor string
Is there an end of line
No
B
Store descriptor string & length
p8

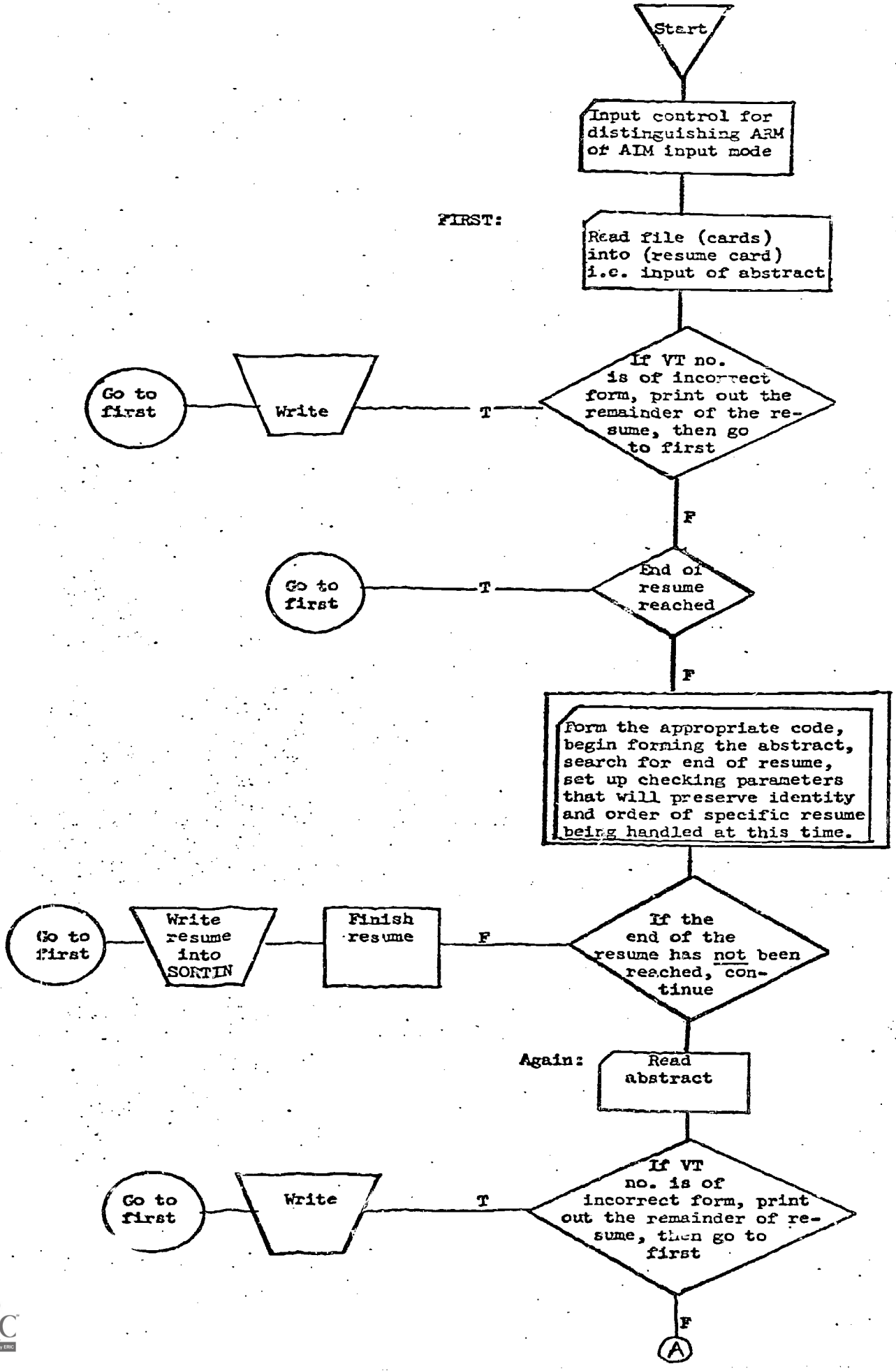
MASTER


p8
Is there an identifier string
No
Is there a VT card
No
Print missing VT card
Yes
Yes
Is there an end of line
No
B
Write record on master tape
A
Yes
Store identifiers & length

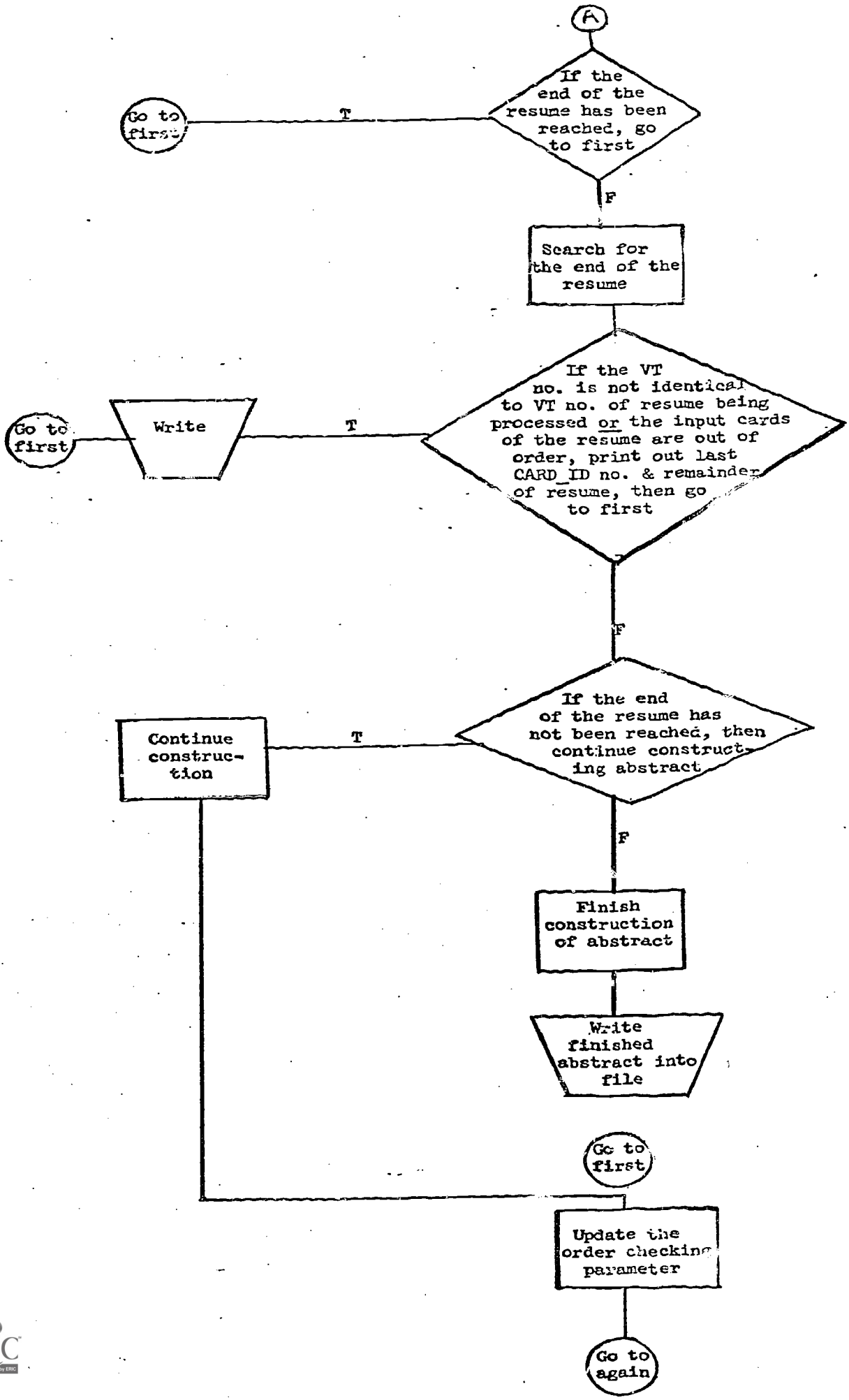

START
Read Cards
Is VT number Correct
No
Print VT number wrong
Read rest of this document
1
Yes
Is line number correct
No
Print line number wrong
Yes
Add to string for this line
Is this end of this section
No
Yes
Was there any information
Yes
Go to Next Section
Return
No
Is lack of information
Yes
No
Print missing information

ABSTRACT

Create Abstract Records

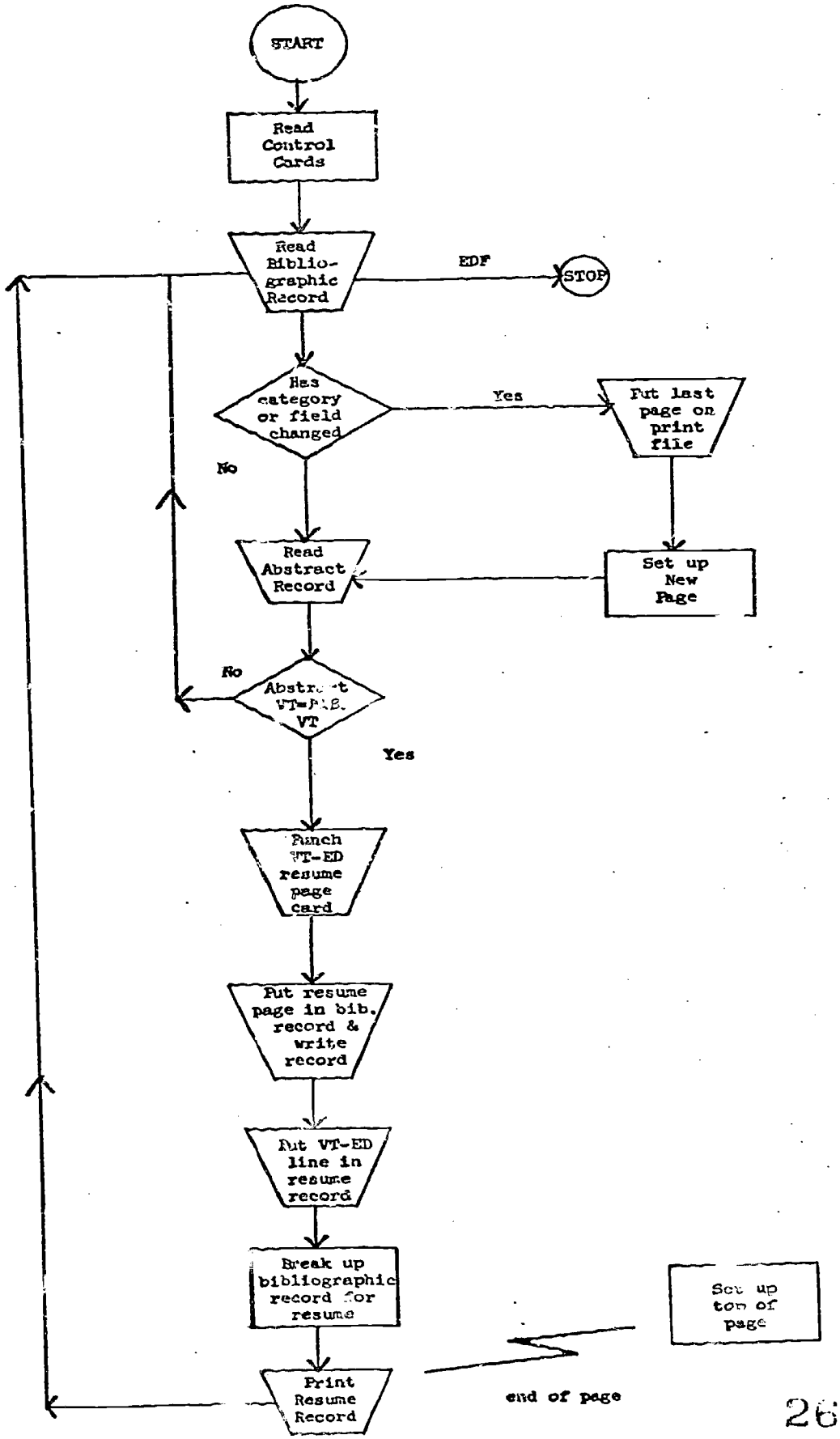


Create Abstract Records (cont'd.)



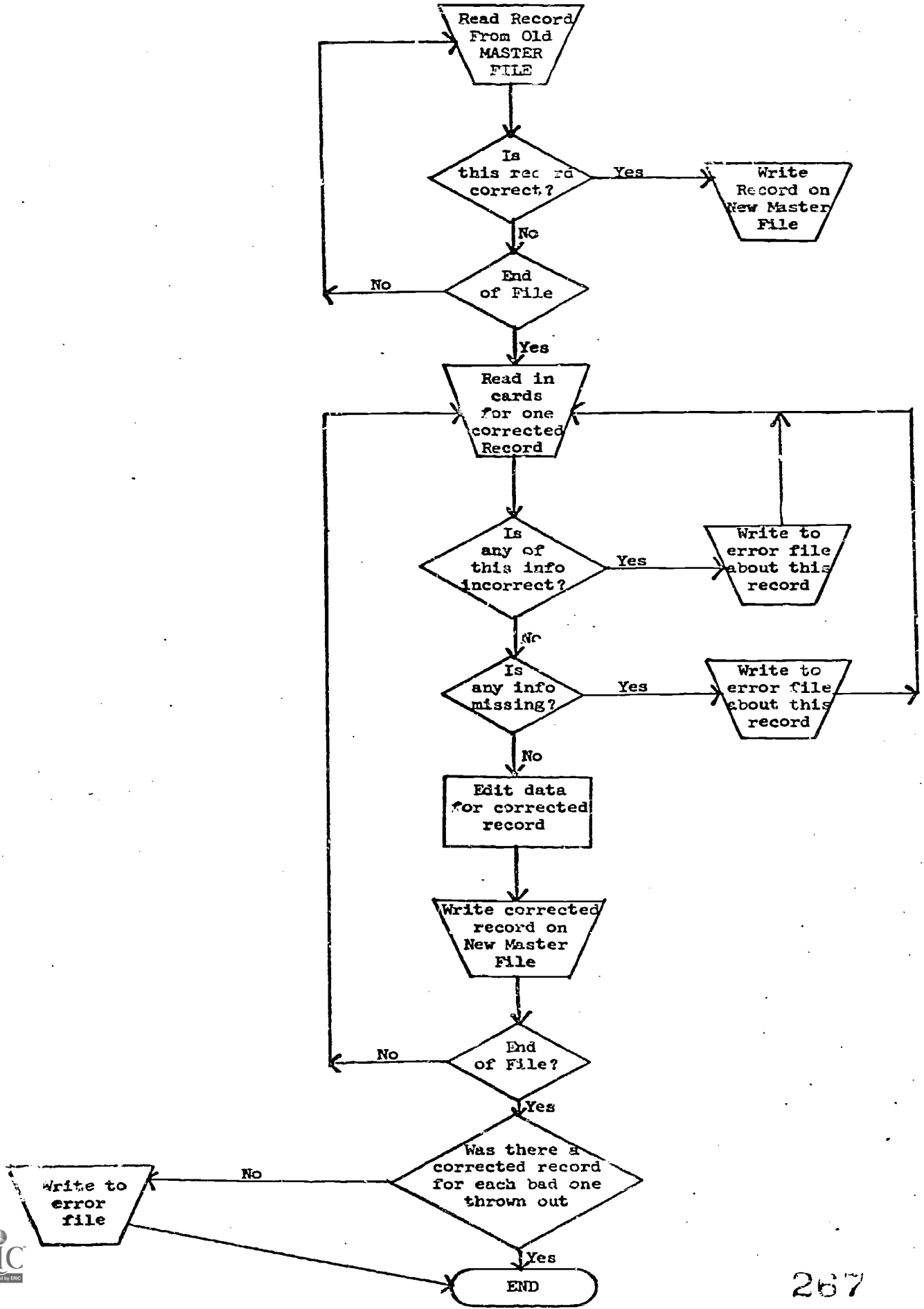
RESUME PRINT

RESUME PRINT

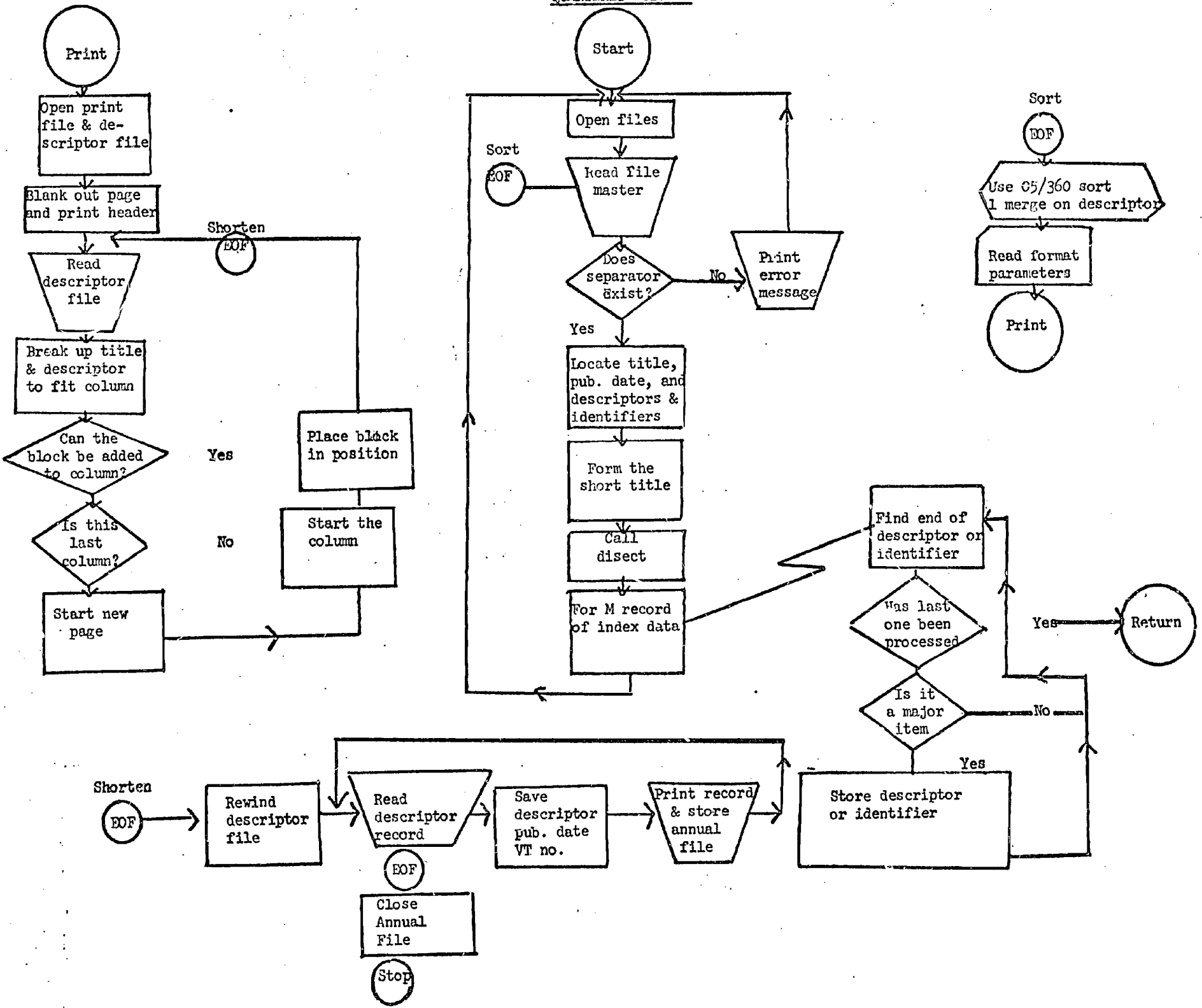


MASTER EDIT

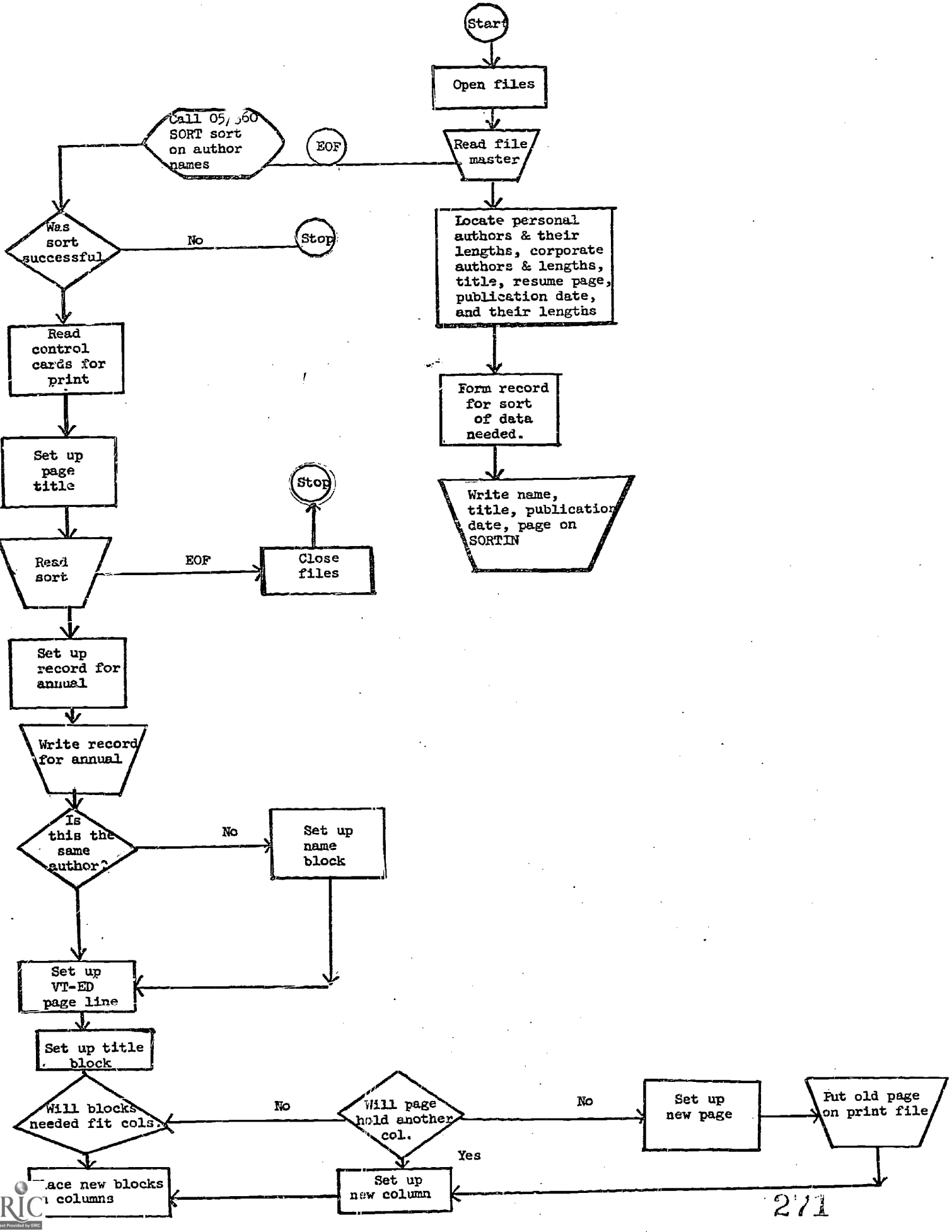
MASTER EDIT


Read Record From Old MASTER FILE
Is this record correct?
Yes
Write Record on New Master File
No
End of File
No
Yes
Read in cards for one corrected Record
Is any of this info incorrect?
Yes
Write to error file about this record
No
Is any info missing?
Yes
Write to error file about this record
No
Edit data for corrected record
Write corrected record on New Master File
End of File?
No
Yes
Was there a corrected record for each bad one thrown out
No
Write to error file
Yes
END

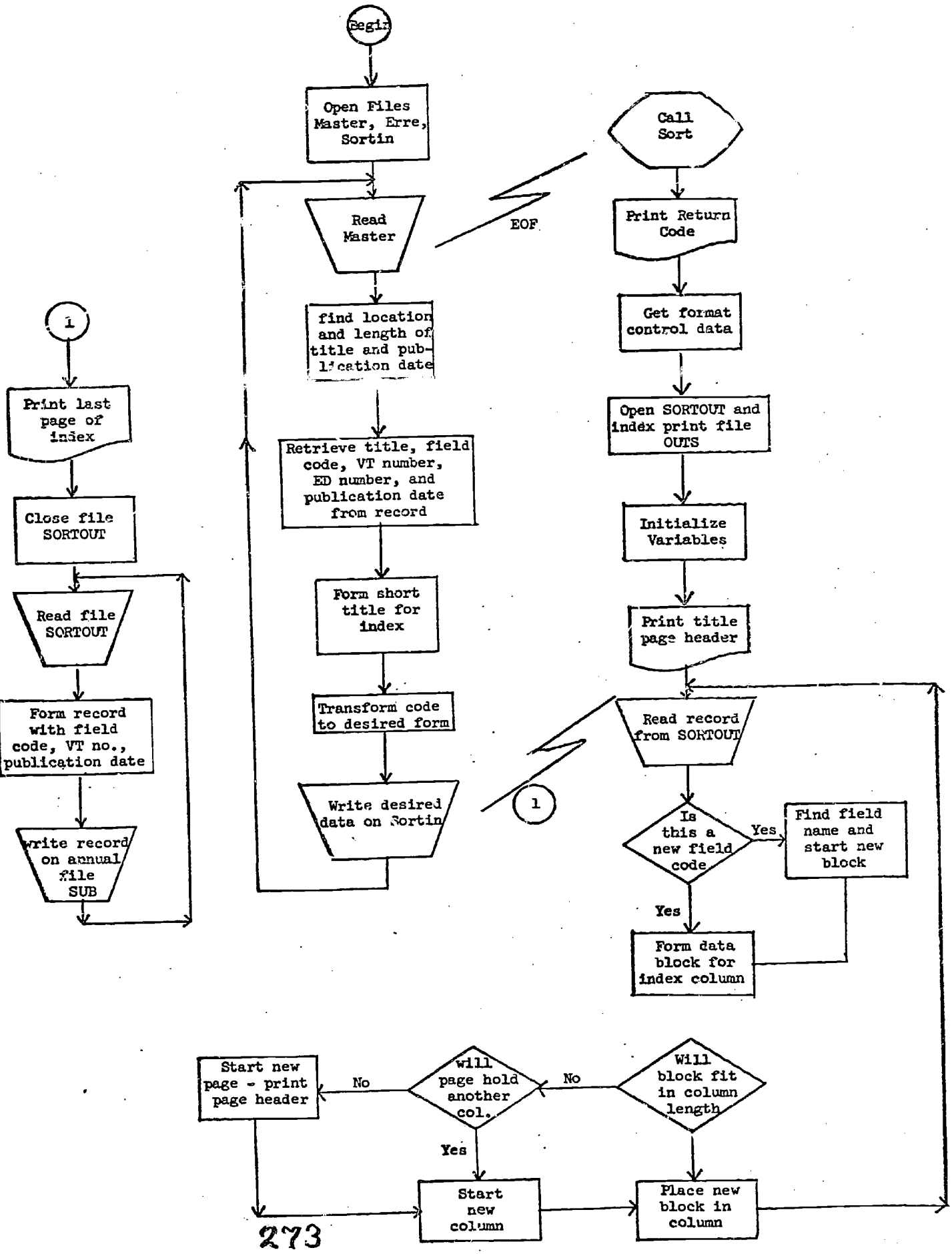
SUBJECT QUARTERLY


Print
Open print file & descriptor file
Blank out page and print header
Shorten
EOF
Read descriptor file
Break up title & descriptor to fit column
Can the block be added to column?
Yes
Place block in position
Is this last column?
No
Start the column
Start new page
Start
Open files
Sort
EOF
Read file master
Does separator exist?
No
Print error message
Yes
Locate title, pub. date, and descriptors & identifiers
Form the short title
Call disect
For M record of index data
Sort
EOF
Use C5/360 sort 1 merge on descriptor
Read format parameters
Print
Find end of descriptor or identifier
Was last one been processed
Yes
Return
Is it a major item
No
Yes
Store descriptor or identifier
Shorten
EOF
Rewind descriptor file
Read descriptor record
EOF
Close Annual File
Stop
Save descriptor pub. date VT no.
Print record & store annual file

AUTHOR QUARTERLY

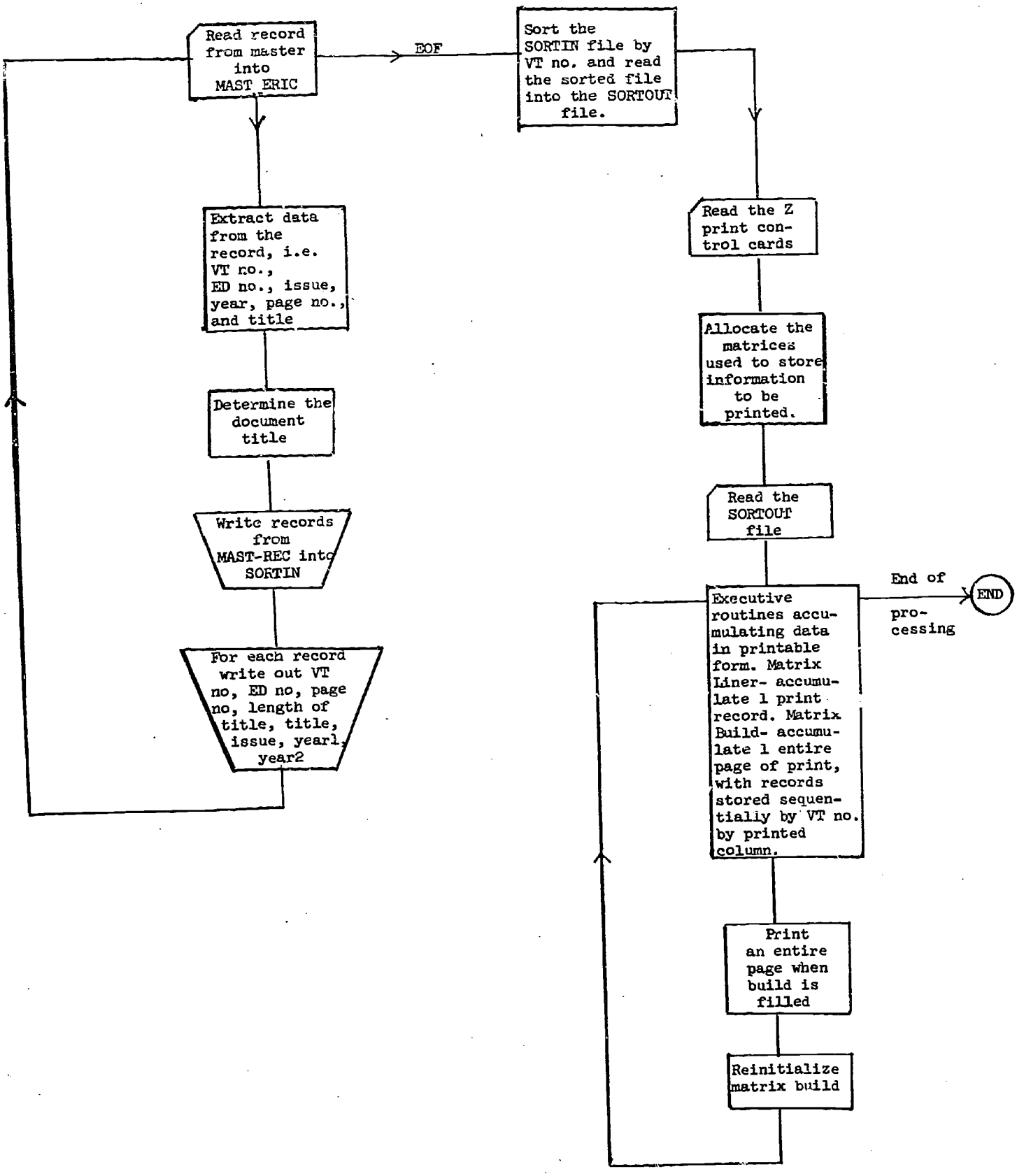

Start
Open files
Read file master
EOF
Call 05/360 SORT sort on author names
Was sort successful
No
Stop
Locate personal authors & their lengths, corporate authors & lengths, title, resume page, publication date, and their lengths
Form record for sort of data needed.
Write name, title, publication date, page on SORTIN
Read control cards for print
Set up page title
Stop
Read sort
EOF
Close files
Set up record for annual
Write record for annual
Is this the same author?
No
Set up name block
Set up VT-ED page line
Set up title block
Will blocks needed fit cols.
No
Will page hold another col.
No
Set up new page
Put old page on print file
Yes
Set up new column
ace new blocks columns

SUPPORTING SERVICES QUARTERLY


Begin
Open Files Master, Erre, Sortin
Read Master
EOF
find location and length of title and publication date
Retrieve title, field code, VT number, ED number, and publication date from record
Form short title for index
Transform code to desired form
Write desired data on Sortin
1
Call Sort
Print Return Code
Get format control data
Open SORTOUT and index print file OUTS
Initialize Variables
Print title page header
Read record from SORTOUT
Is this a new field code
Yes
Find field name and start new block
Yes
Form data block for index column
will page hold another col.
No
Start new page - print page header
Will block fit in column length
No
Yes
Start new column
Place new block in column
1
Print last page of index
Close file SORTOUT
Read file SORTOUT
Form record with field code, VT no., publication date
Write record on annual file SUB

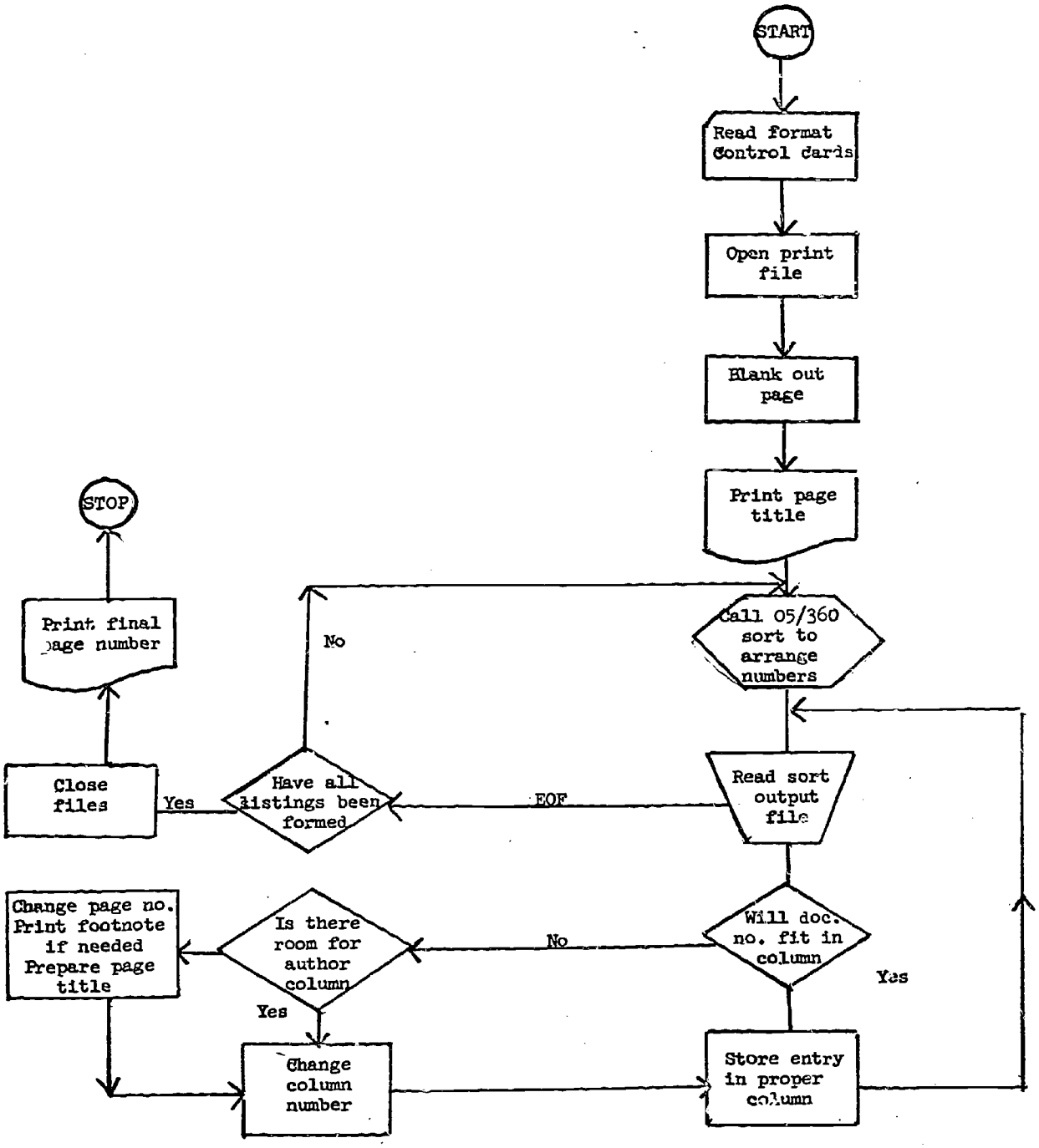
DOCUMENT NUMBER QUARTERLY

Document Number Quarterly



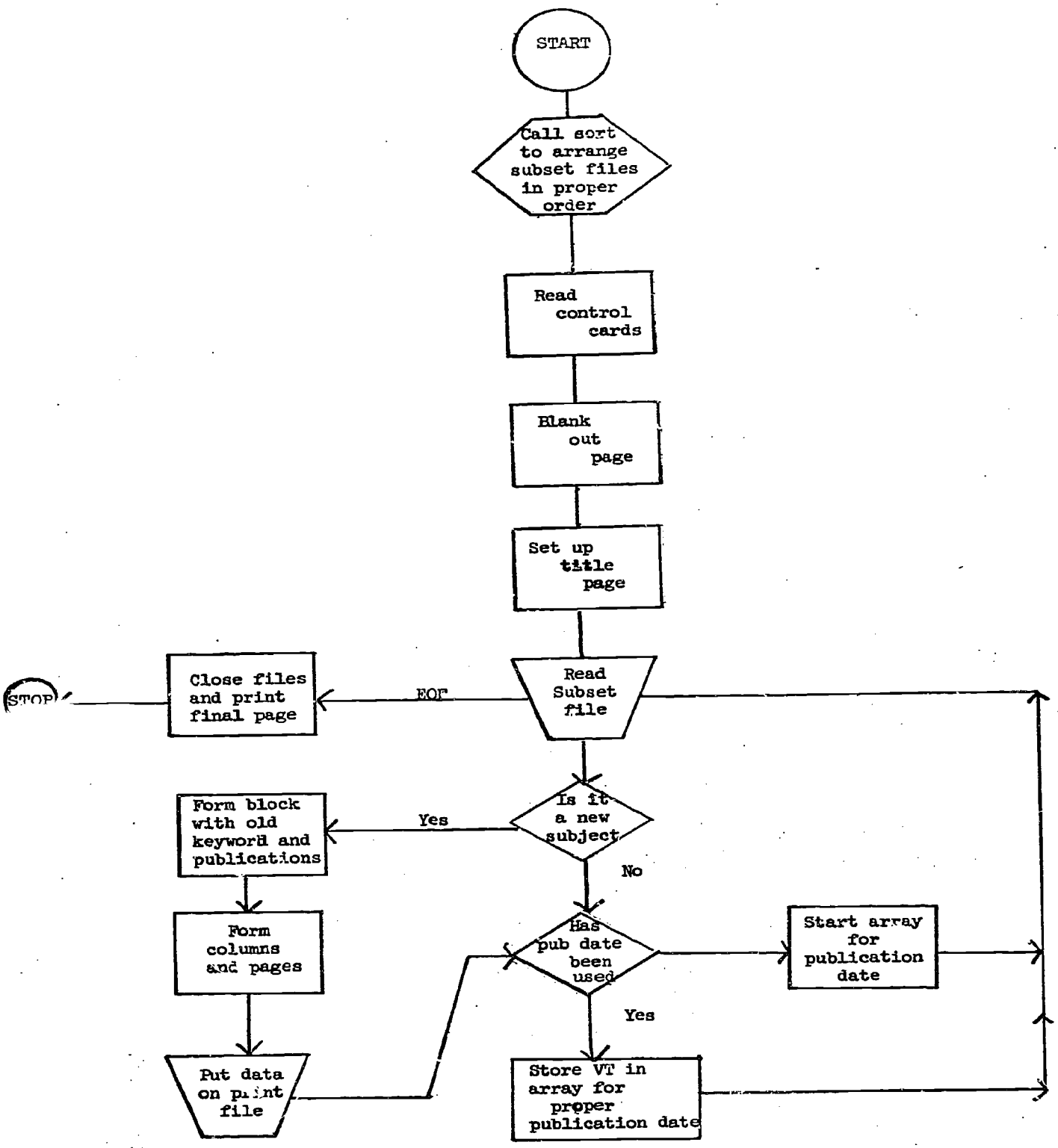
CONVERSION OF DOCUMENT NUMBER

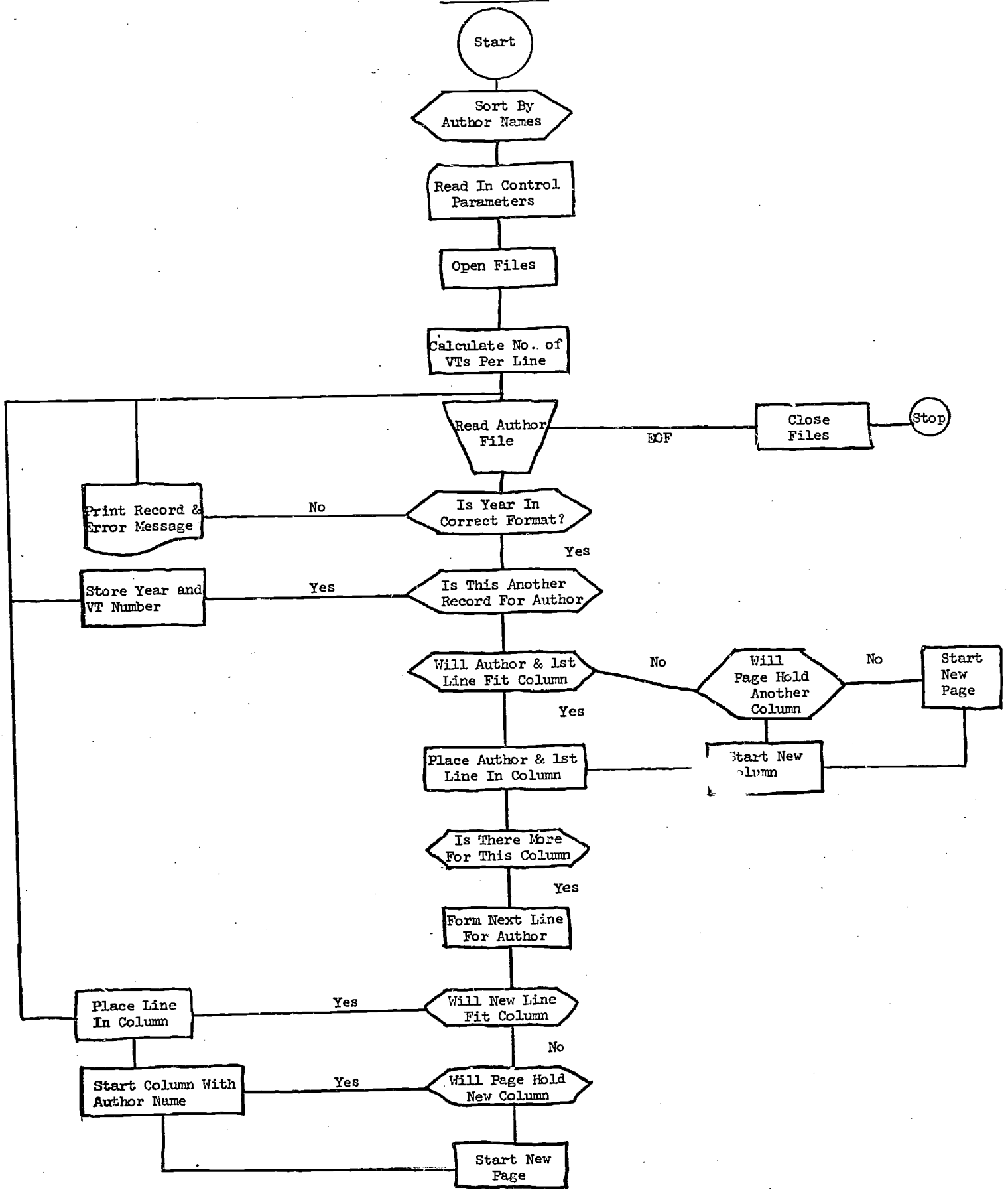
CONVERSION OF DOCUMENT NUMBER



SUBJECT ANNUAL

SUBJECT ANNUAL

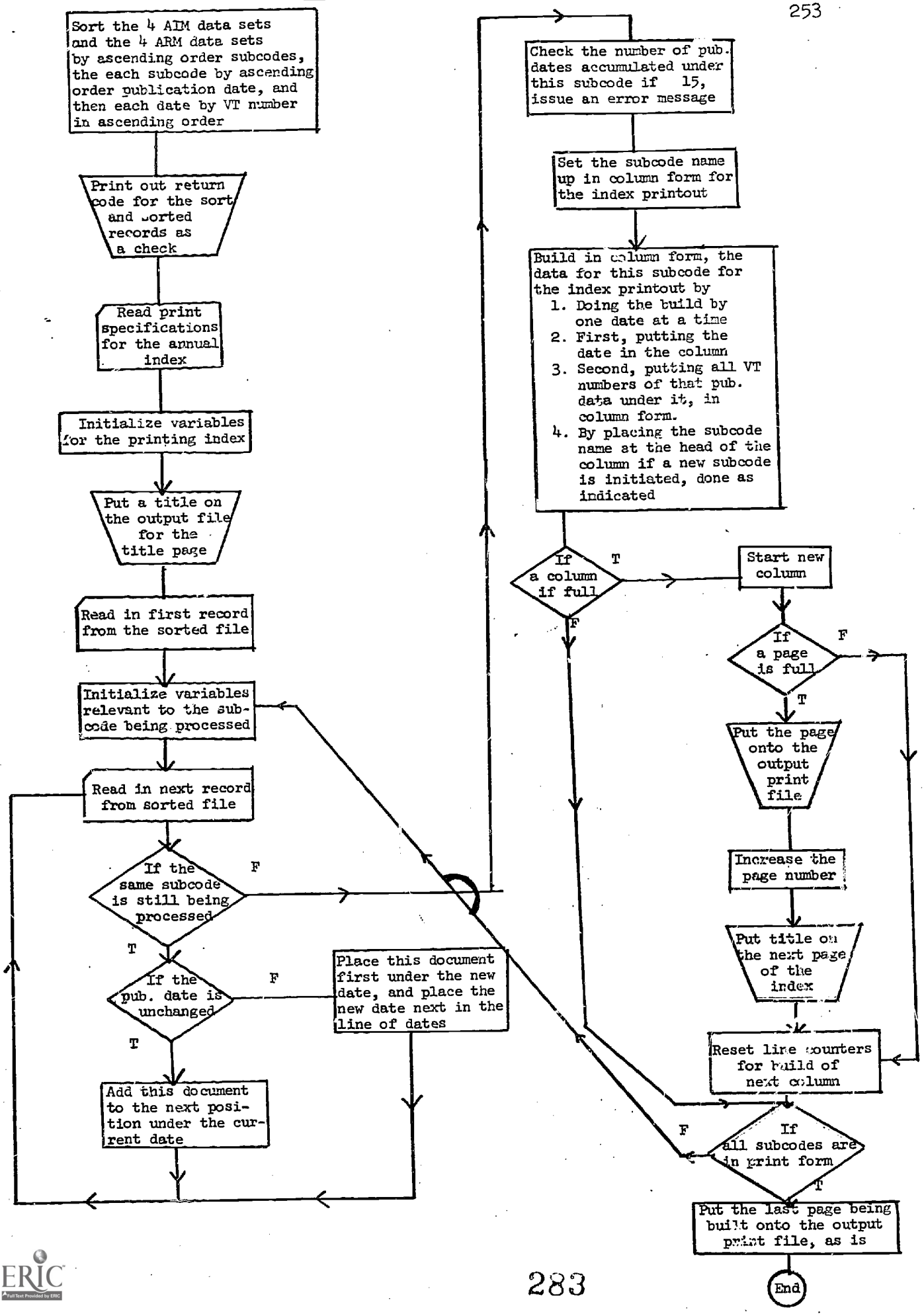



Start
Sort By Author Names
Read In Control Parameters
Open Files
Calculate No. of VTs Per Line
Read Author File
EOF
Close Files
Stop
Print Record & Error Message
No
Is Year In Correct Format?
Yes
Store Year and VT Number
Yes
Is This Another Record For Author
Will Author & 1st Line Fit Column
No
Will Page Hold Another Column
No
Start New Page
Yes
Place Author & 1st Line In Column
Start New Column
Is There More For This Column
Yes
Form Next Line For Author
Place Line In Column
Yes
Will New Line Fit Column
No
Start Column With Author Name
Yes
Will Page Hold New Column
Start New Page

AUTHOR ANNUAL

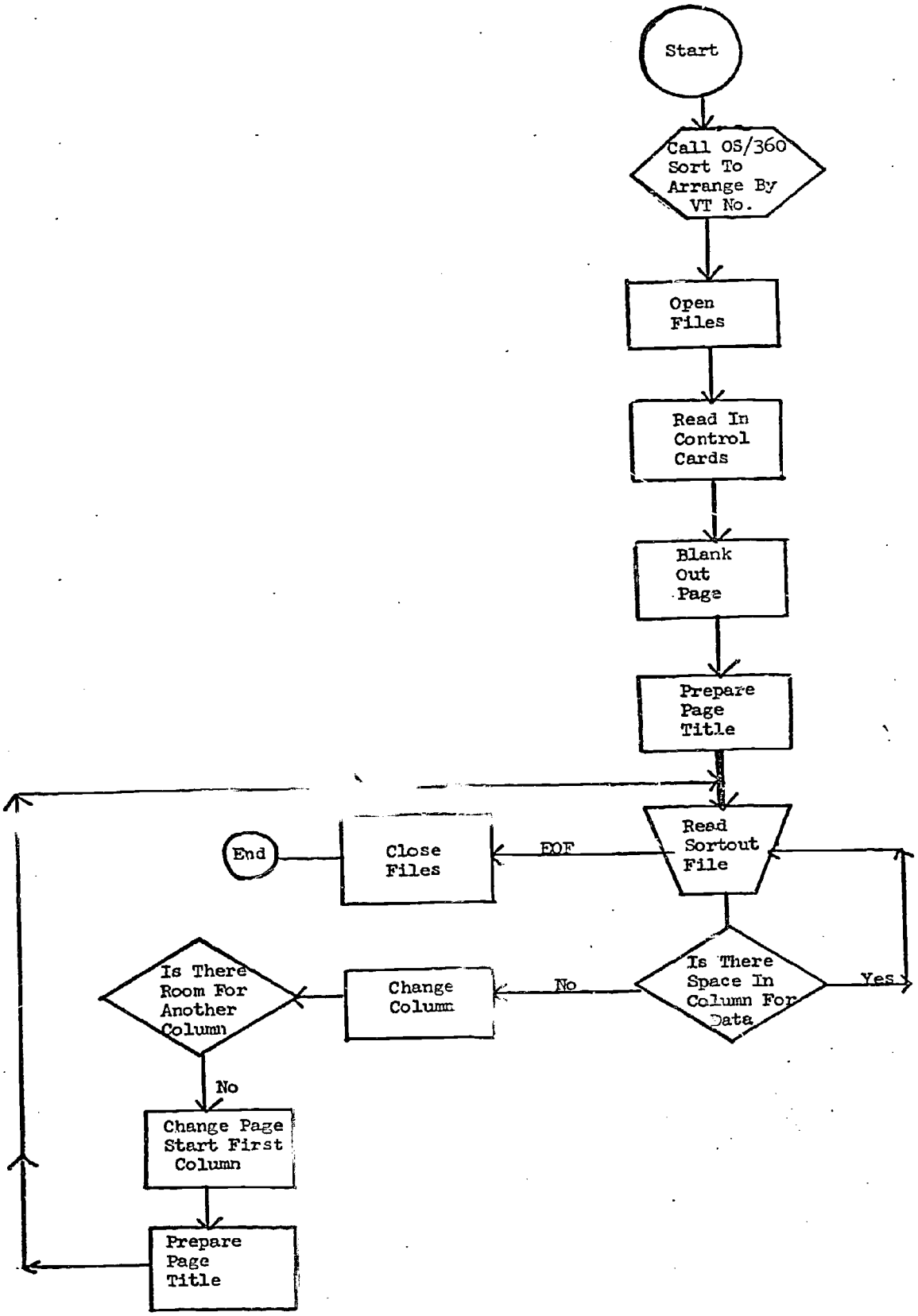
SUPPORTING SERVICES ANNUAL

FLOW CHART FOR ANNUAL SUPPORTING SERVICES INDEX PROGRAM



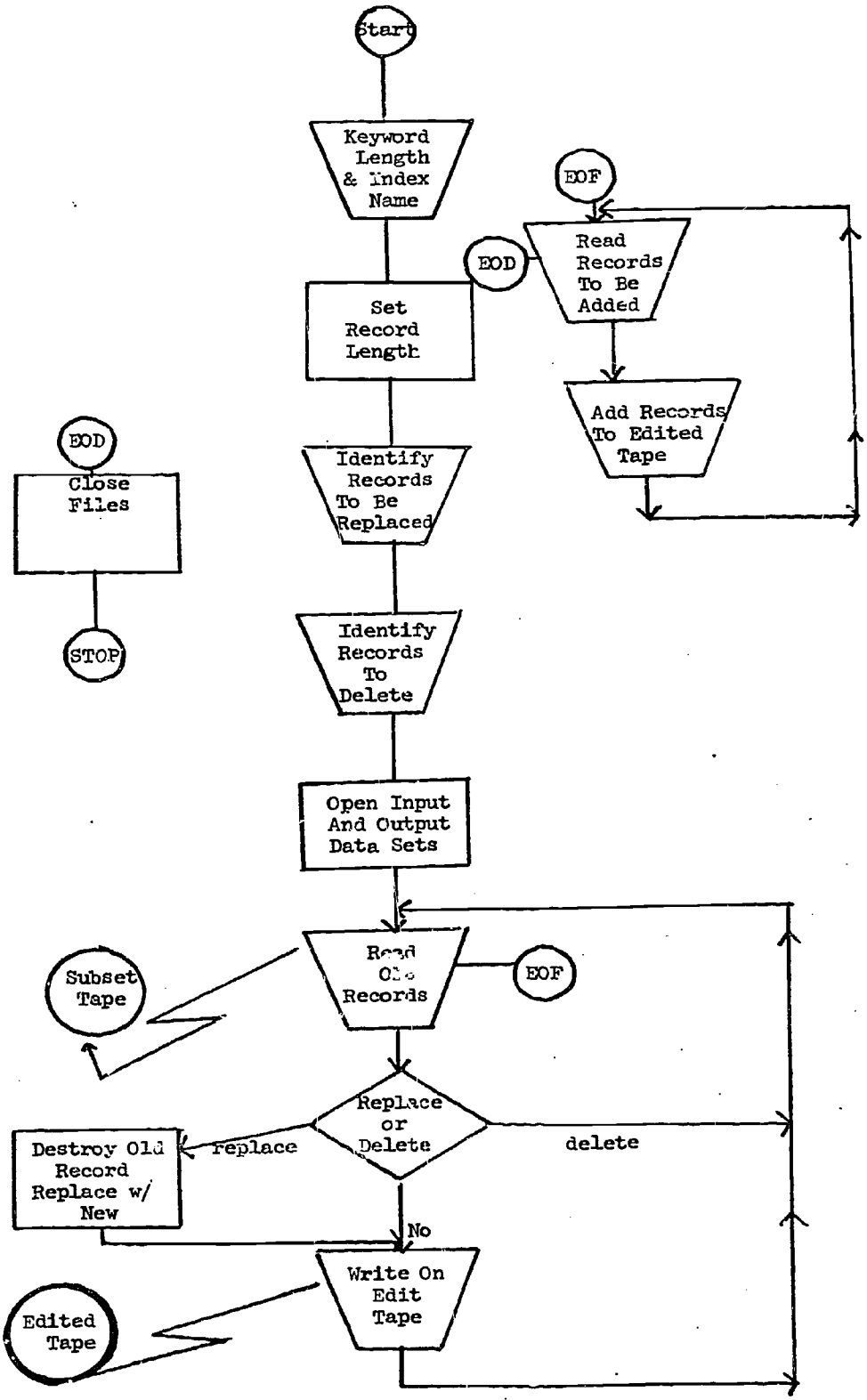
DOCUMENT NUMBER ANNUAL

DOCUMENT NUMBER ANNUAL



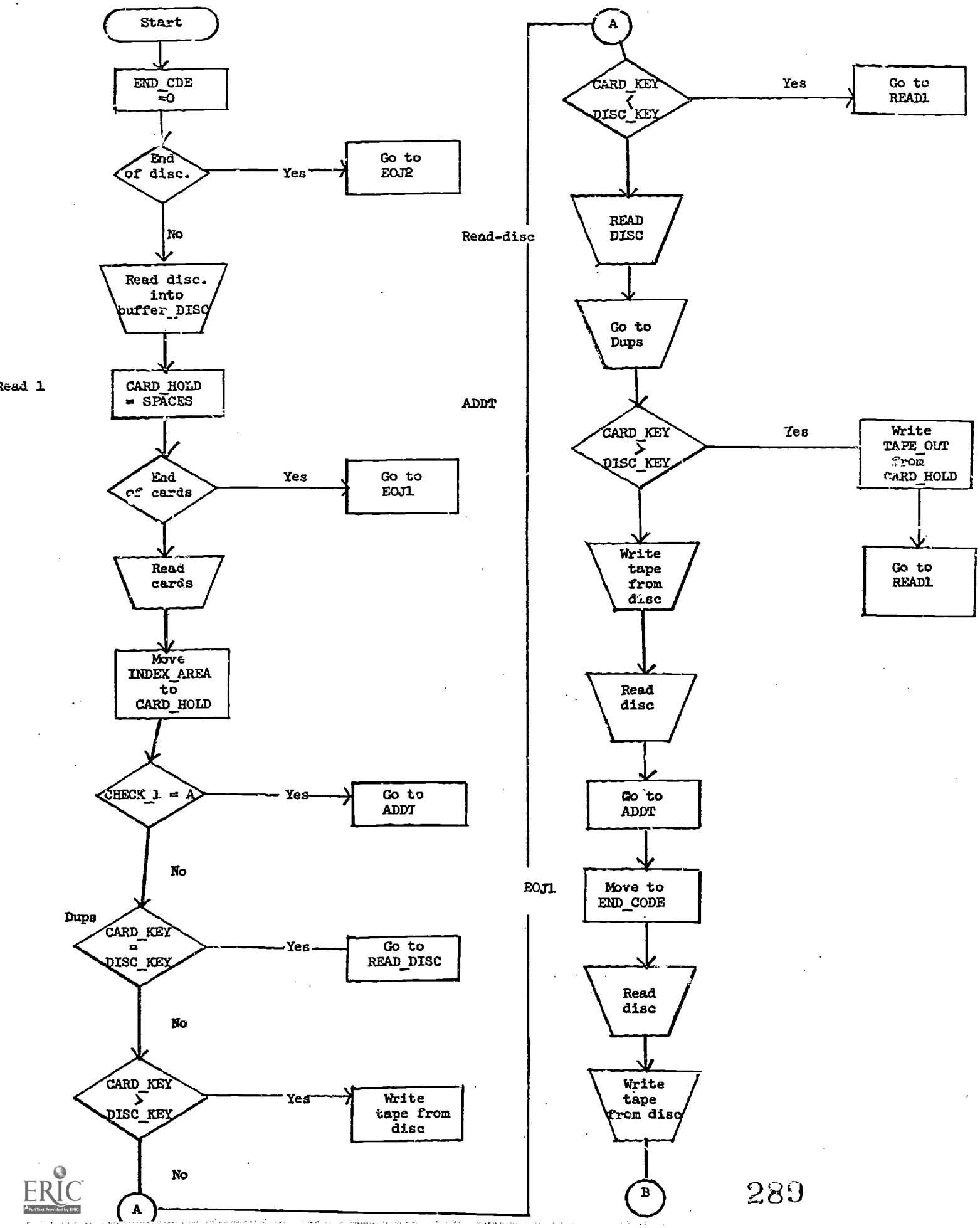
EDIT OF SUBJECT, AUTHOR, AND SUPPORTING SERVICES

EDIT OF SUBJECT, AUTHOR, AND SUPPORTING SERVICE

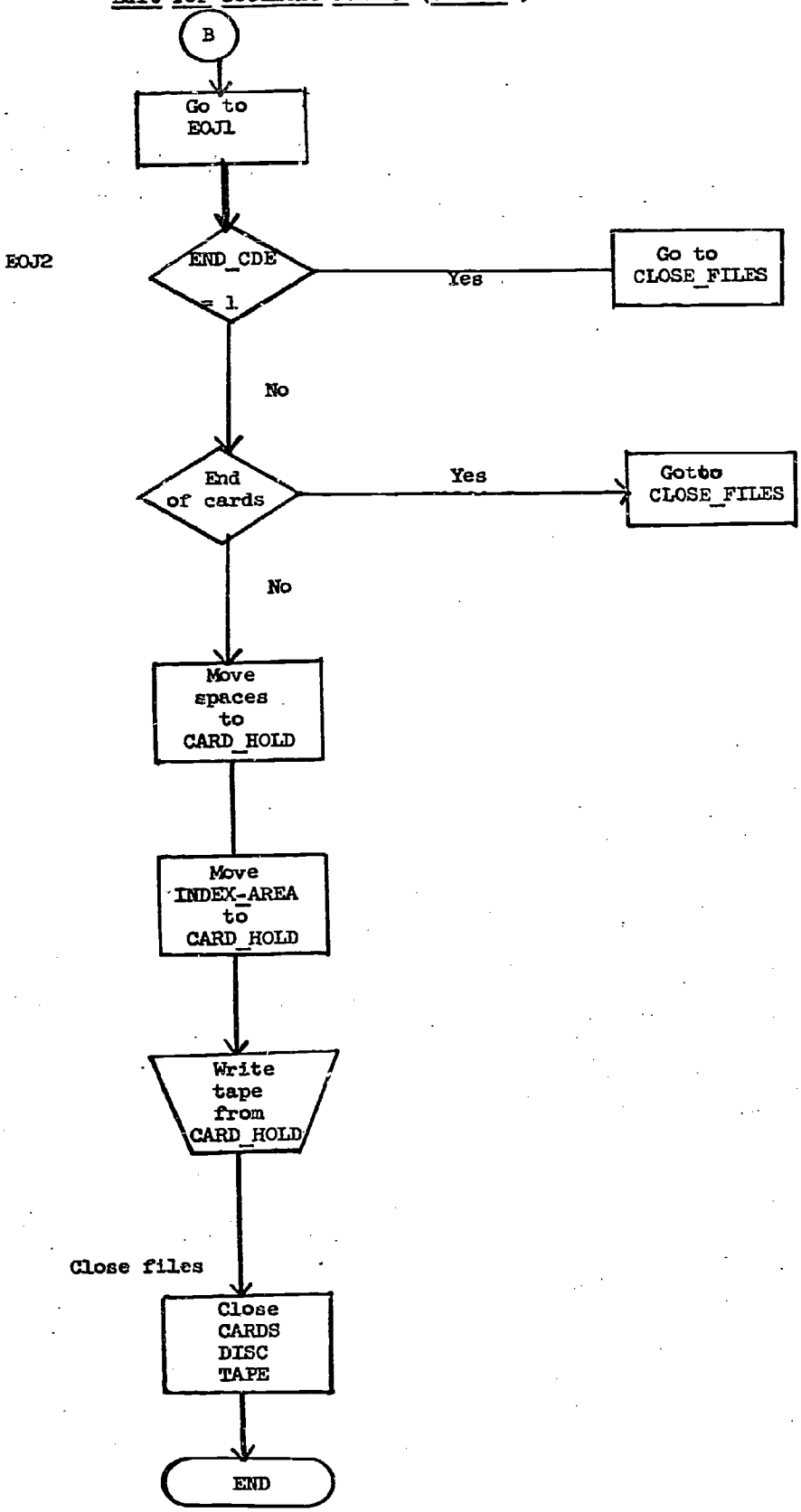


DOCUMENT NUMBER EDIT

Edit for Document Number

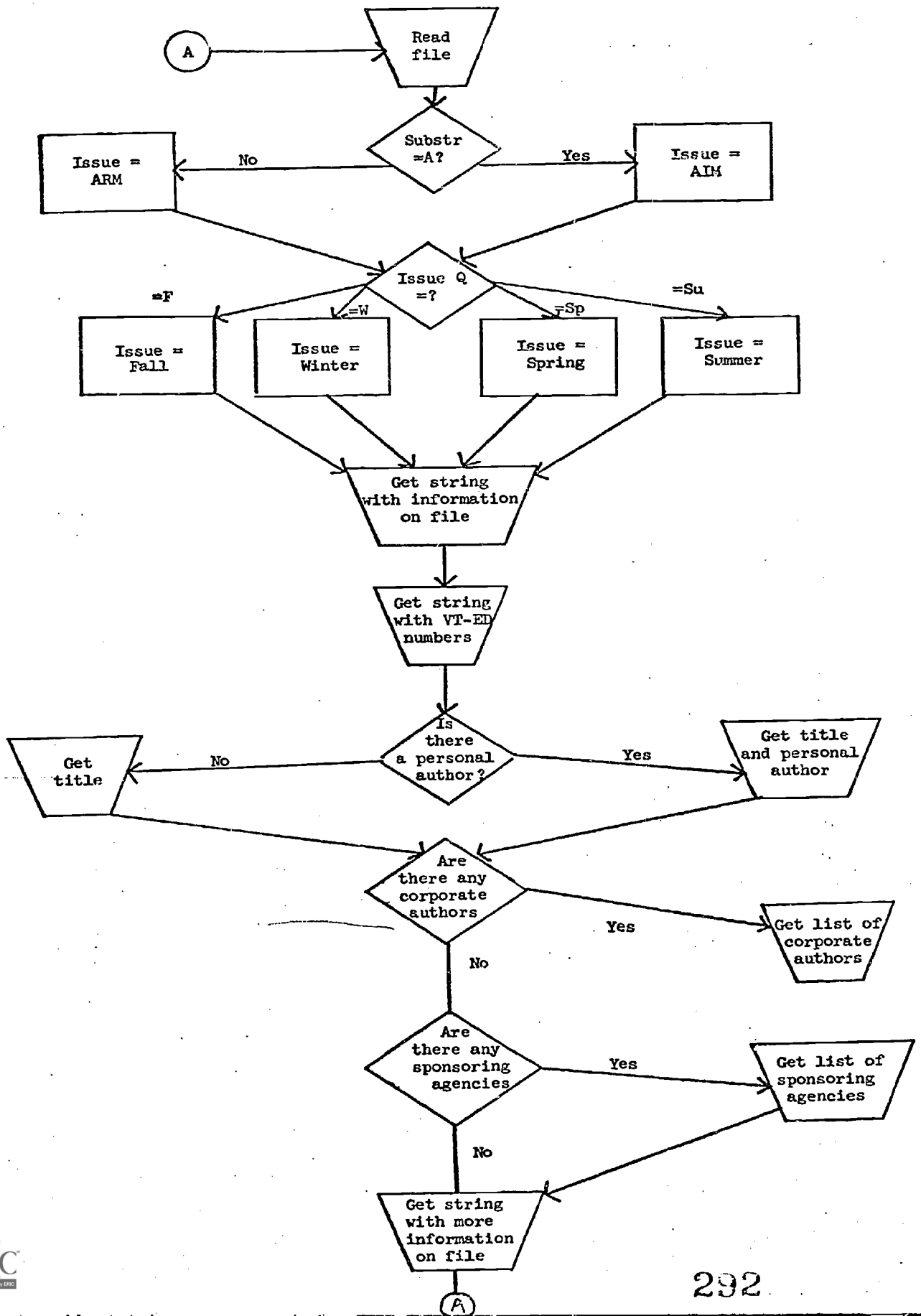


Edit for Document Number (cont'd.)

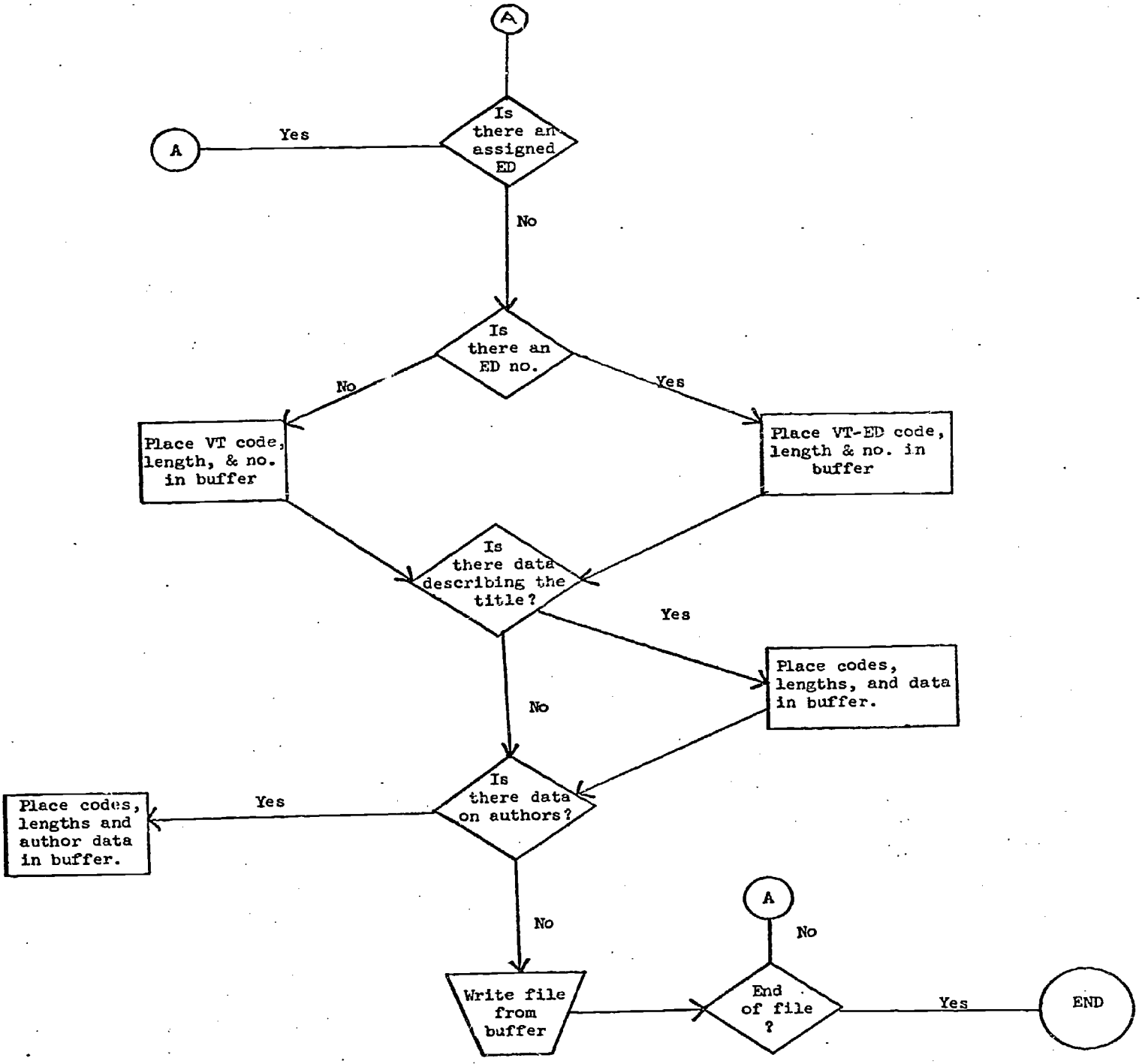


RIE FORMAT

RIE Format


A
Read file
Substr =A?
No
Yes
Issue = ARM
Issue = AIM
Issue Q =?
=F
=W
=Sp
=Su
Issue = Fall
Issue = Winter
Issue = Spring
Issue = Summer
Get string with information on file
Get string with VT-ED numbers
Is there a personal author?
No
Yes
Get title
Get title and personal author
Are there any corporate authors
Yes
Get list of corporate authors
No
Are there any sponsoring agencies
Yes
Get list of sponsoring agencies
No
Get string with more information on file
A

RIE Format (cont'd.)



APPENDIX G

PAGE SAMPLES

	page
RESUME SECTION	265
QUARTERLY SUBJECT INDEX	268
QUARTERLY AUTHOR INDEX	271
QUARTERLY SUPPORTING SERVICE INDEX	274
QUARTERLY DOCUMENT NUMBER INDEX	277
CONVERSION OF DOCUMENT NUMBER INDEX	280
ANNUAL SUBJECT INDEX	282
ANNUAL AUTHOR INDEX	285
ANNUAL SUPPORTING SERVICE INDEX	288
ANNUAL DOCUMENT NUMBER INDEX	291

RESUME SECTION

VT 010 484
Siegfried, Larry R.
Source Unit in Feeds and Feeding.

Colorado State Univ., Ft. Collins. Agricultural Education Section
Colorado State Board for Community Colleges and Occupational Education, Denver
MF AVAILABLE IN VT-ERIC SET.
Agricultural Education Section, Department of Vocational Education, Colorado State University, Fort Collins, Colorado 80521 ($1.00).
PUB DATE - 69 41p.

DESCRIPTORS - *RESOURCE UNITS; RESOURCE MATERIALS; *UNIT PLAN; *VOCATIONAL AGRICULTURE; *AGRICULTURAL PRODUCTION; ANIMAL SCIENCE; *NUTRITION; LIVESTOCK; SECONDARY GRADES

ABSTRACT - Prepared by a graduate assistant in agricultural education, this source unit may be used by vocational agriculture teachers in planning and presenting a unit in feeds and feeding. The 13 sub-units, providing objectives, motivation techniques, study guides, references, and suggested teaching plans, cover: (1) nutrients necessary for livestock, (2) chemical composition and classification of feeds, (3) digestive systems of ruminants and non-ruminants, (4) digestion and adsorption of feeds by livestock, (5) feed preparation techniques, (6) role of proteins, carbohydrates, fats, minerals, and vitamins in livestock feeding, (7) feed additives, (8) balancing rations, and (9) computing lowest-cost balanced rations. (AW)

VT 010 527
Tree Identification: Western Pines.

California State Polytechnic Coll., San Lnis Obispo. Vocational Education Productions
California State Dept. of Education, Sacramento. Vocational Education Section
MF AVAILABLE IN VT-ERIC SET.
Vocational Education Productions, California State Polytechnic College, San Luis Obispo, California 93401 ($2.00, Manual and 20 slides available for $6.50).
PUB DATE - 69 21p.

DESCRIPTORS - *GUIDES; *AGRICULTURAL EDUCATION; *IDENTIFICATION; *TREES; *FORESTRY; NATURAL RESOURCES; CLASSIFICATION; TAXONOMY

ABSTRACT - This multi-media kit, for use in agricultural education classes studying tree identification, consists of the following: (1) 10 slides of pines of commercial significance in the West, with the common and Latin names of the tree superimposed on a combined view of a typical specimen and a close-up view of the trunk and needles or leaves, (2) 10 slides identical to the ones identified, but without the names, to be used for testing purposes, (3) 10 transparency masters, one for each of the 10 species pictured on the slides, for use in producing overhead transparencies or other instructional materials, (4) short descriptions of the 10 pictured species, and (5) suggested identification procedures. The 10 Species of western pines covered are: (1) Douglas Fir, (2) Engelmann Spruce, (3) Idaho White Pine, (4) Incense Cedar, (5) Larch, (6) Lodgepole Pine, (7) Ponderosa Pine, (8) Sugar Pine, (9) Western Red Cedar, and (10) White Fir. The slide set may be obtained from the source of availability indicated for this document. (AW)

VT 010 528
Careers in Animal Industry.

California State Polytechnic Coll., San Luis Obispo. Vocational Education Productions
California State Dept. of Education, Sacramento. Vocational Education Section
DOCUMENT NOT AVAILABLE FROM EDRS.
Vocational Education Productions, California State Polytechnic College, San Luis Obispo, California 93401 ($.50; filmstrip, script, and taped narration $7.95).
PUB DATE - 69 15p.

DESCRIPTORS - *AUDIOVISUAL AIDS; *OCCUPATIONAL INFORMATION; *AGRICULTURAL OCCUPATIONS; *OFF FARM AGRICULTURAL OCCUPATIONS
IDENTIFIERS - *ANIMAL INDUSTRY

ABSTRACT - This illustrated companion script to a 50-frame filmstrip was developed by departments in the School of Agriculture, California State Polytechnic College to provide information on the history, progress, and variety of occupations in the animal industry. The film also discusses briefly the education needed and opportunities for advancement in animal industry occupations. (CH)

DESC[illegible]TORS - *TEACHING GUIDES; *DISTRIBUTIVE E[illegible]ATION; *TEACHING TECHNIQUES; SALESMANSHIP; MERCHANDISING; COMMUNICATION SKIL[illegible] CARE[illegible] PLANNING; SECONDARY GRADES

ABSTRACT - The teacher's guide is designed to a[illegible] the teacher-coordinator in planning individual and group activities for students in [illegible]stributive Education I and II. The teaching techniques suggested include case stud[illegible]s, structured reports, and assignments related to class study, the students' jobs, and [illegible]lub contests. Units containing suggested activities for Distributive Education [illegible] are: (1) School and Business Relationships, (2) Sales Development, (3) Busin[illegible]s Communications, and (4) Organizations for Distribution. [illegible]nits containing suggested activities for Distributive Education II are: (1) Career Planning for Profes[illegible]onal Growth, (2) Economics for Distribution, (3) Merchandising, and (4) Busines[illegible] Psychology. Each unit contains teacher information and student forms for the su[illegible]gested activities. (SB)

VT 002 683
Marketing I; Teachers Manual. Distributive Educa[illegible]ion Se[illegible]ies.

North Carolina State Dept. of Public Instruction, Raleigh. Div. of Vocational Education
MF AVAILABLE IN VT-ERIC SET.
PUB DATE - ND 247p.

DESCRIPTORS - *DISTRIBUTIVE EDUCATION; *TEACHING GUIDES; *MARKETING; SALES OCCUPATIONS; SALESMANSHIP; MERCHANDISING; CAREER OPPORTUNITIES; GRADE 11

ABSTRACT - Developed by a committee of high schoo[illegible] teacher-coordinators, this teacher's guide is for use in planning a first-year course in marketing for juniors who expect to enter some form of distribution, but who cannot, for various reasons, participate in a cooperative program of on-the-job training. A sug[illegible]ested time allotment and bibliography are provided for these units: (1) Introduction to Marketing I, (2) Sales Supporting Activities, (3) Salesmanship, (4) Sales Promotion, (5) Business Organization, and (6) Careers in Marketing. Each unit contains a list of materials and equipment needed, objectives, topical outline, and teaching suggestions. Supplementary forms and information for students are appended. A related document is available as VT 002 568 (also in this issue). (SB)

VT 010 256
Russell, Kenneth Lee
The How in Parliamentary Procedure.

Distributive Education Clubs of America
DOCUMENT NOT AVAILABLE FROM EDRS.
The Interstate Printers and Publishers, Inc., 19-27 North Jackson Street, Danville, Illinois 61832.
PUB DATE - 67 65p.

DESCRIPTORS - *DISTRIBUTIVE EDUCATION; *STUDENT ORGANIZATIONS; *CLASSROOM MATERIALS; *COCURRICULAR ACTIVITIES; *MANUALS
IDENTIFIERS - *PARLIAMENTARY PROCEDURES

ABSTRACT - This handbook was prepared by a professor of education for those who participate in group meetings. It is useful both to high school groups such as student councils and to adult civic groups. Topics discussed include (1) rule enforcement, (2) motions, (3) elections, (4) new business, and (5) requests. A student assignment book based on this manual is available as VT 010 257 (also in this issue). (JS)

VT 010 257
The How in Parliamentary Procedure, Student Assignments.

Texas Univ., Austin. Instructional Materials Lab.
DOCUMENT NOT AVAILABLE FROM EDRS.
Instructional Materials Laboratory, Distributive Education Department, Division of Extension, The University of Texas, Austin, Texas 78712.
PUB DATE - Jun69 29p.; Based on "The How in Parliamentary Procedure" by Kenneth L. Russell, The Interstate Printers and Publishers, Inc., Danville, Illinois, 1967.

DESCRIPTORS - *DISTRIBUTIVE EDUCATION; MANUALS; *STUDENT PROJECTS; *WORKBOOKS; *STUDY GUIDES; SECONDARY SCHOOLS; *CLASSROOM MATERIALS
IDENTIFIERS - *PARLIAMENTARY PROCEDURES

ABSTRACT - This student manual was published by a university materials laboratory to distributive education students the fundamentals of parliamentary procedure.

QUARTERLY SUBJECT INDEX

SUBJECT AND IDENTIFIER INDEX

PAGE

ABSTRACTS
VT 010 767 ED 037 582 81
Abstracts of Instructional Materials for Student Use in Vocational Cooperative Education.

ACCIDENT PREVENTION
VT 010 904 17
Farm Machinery Safety for the Institutional Farm.

ADMINISTRATOR GUIDES
VT 002 685 58
Trade Preparatory Training Planning Guide, Carpentry.
VT 010 615 22
Vocational Office Education in Florida High Schools.
VT 010 773 23
Administrative Handbook for Business and Distributive Education.
VT 011 216 ED 040 291 78
Applying the Cooperative Plan of Instruction to Manpower Programs. A Handbook for Supervisors and Coordinators.

ADULT EDUCATION
VT 010 245 21
Introduction to Automatic Data Processing.
VT 011 140 ED 039 365 24
Information Communication Occupations. A Suggested Curriculum Guide.
VT 011 169 26
Office Machines Resource Materials for Adult Programs in Business and Distributive Education.

ADULT FARMER EDUCATION
VT 002 841 6
Livestock Feeds and Feeding.

ADULT VOCATIONAL EDUCATION
VT 008 757 62
Welding and Flame-Cutting Processes and Practices.
VT 010 570 34
Franchise Merchandising. A Suggested Adult Course Outline.
VT 011 220 42
Oral Histology.

ADVERTISING LAYOUT
VT 010 412 32
Advertising.Layout and Copy.

AGRICULTURAL CHEMICAL OCCUPATIONS
VT 001 346 5
Agricultural Chemicals.
VT 008 503 7
Agricultural Chemicals Employee.

PAGE

AGRICULTURAL EDUCATION
VT 000 382 3
Ornamental Horticulture for Vocational Agriculture in Alabama.
VT 007 073 7
A Handbook for Student Teachers and Supervising Teachers in Vocational Agriculture.
VT 010 362 9
Legal Principles for Farm Managers.
VT 010 363 9
Agricultural Cooperatives.
VT 010 527 12
Tree Identification: Western Pines.
VT 010 754 15
Fundamentals of Service: Power Trains.
VT 010 755 16
Fundamentals of Service: Hydraulics.
VT 010 756 16
Fundamentals of Service: Engines.
VT 010 757 16
Fundamentals of Service: Electrical Systems.
VT 010 968 ED 039 346 18
Farm Crop Production Technology: Field and Forage Crop and Fruit and Vine Production Options. A Suggested Curriculum.

AGRICULTURAL ENGINEERING
VT 001 571 5
Vocational Agriculture Course of Study Guide.
VT 010 251 7
Cold Metal Work in Vocational Agriculture.
VT 010 255 8
Overhead Projection; Transparency Masters for Vocational Agriculture (Including Supplements I-III).
VT 010 259 8
Course Guide for Basic Agricultural Mechanics.
VT 010 360 8
The Internal Combustion Engine.
VT 010 754 15
Fundamentals of Service: Power Trains.
VT 010 755 16
Fundamentals of Service: Hydraulics.
VT 010 756 16
Fundamentals of Service: Engines.
VT 010 757 16
Fundamentals of Service: Electrical Systems.
VT 010 765 ED 038 527 17
Agricultural Equipment Technology: A Suggested 2-Year Post High School Curriculum.

SUBJECT AND IDENTIFIER INDEX

PAGE

VT 010 235 69
Training Recommendations: Scaffolders.
VT 010 583 74
Building Maintenance, A Study Guide.

BUSINESS EDUCATION
VT 001 873 20
Education for Business.
VT 001 881 20
Bookkeeping I, Syllabus.
VT 010 114 20
Supplementary Materials Handbook for Intensive Business and Office Education.
VT 010 465 21
Data Processing, A Course of Study.
VT 010 570 34
Franchise Merchandising. A Suggested Adult Course Outline.
VT 010 773 23
Administrative Handbook for Business and Distributive Education.
VT 010 774 23
Bookkeeping Motivation Devices.
VT 010 775 23
Typewriting.
VT 010 776 24
Your Business Teaching.
VT 011 145 25
Typewriting for Business; Typewriting by a Whole (Gestalt) Method.

CALCULUS
VT 010 406 80
Calculus For Technicians.

CARBURETOR AIR CLEANERS
VT 010 638 76
Servicing Carburetor Air Cleaners.

CAREER GUIDANCE SERIES
VT 000 239 48
Industrial Arts Supplement.

CAREER PLANNING
VT 001 359 48
Industrial Arts.

CENTRAL SERVICE AIDES
VT 011 073 41
Guidelines for the Training of Central Service Aides.

CHILD CARE
VT 008 419 60
Nursery Assistants. Curriculum Guides in Trade and Industrial Education.

CHILD CARE OCCUPATIONS
VT 008 419 60
Nursery Assistants. Curriculum Guides in Trade and Industrial Education.

PAGE

CIVIL ENGINEERING
VT 002 645 56
Civil Engineering Technology, Field Book.
VT 002 656 56
Civil Engineering Technology.

CLASSROOM MATERIALS
VT 010 256 29
The How in Parliamentary Procedure.
VT 010 257 29
The How in Parliamentary Procedure, Student Assignments.
VT 011 143 25
Timed Writing, No. 1-7.
VT 011 145 25
Typewriting for Business; Typewriting by a Whole (Gestalt) Method.

CLOTHING INSTRUCTION
VT 008 421 61
Dressmaking.

COCURRICULAR ACTIVITIES
VT 010 256 29
The How in Parliamentary Procedure.

COLLEGE BOUND STUDENTS
VT 009 288 51
A Technical Drawing Course for College-Preparatory Students.

COMMUNICATION OCCUPATIONS
VT 011 140 ED 039 365 24
Information Communication Occupations. A Suggested Curriculum Guide.

COMMUNICATIONS
VT 002 661 58
Trade Preparatory Training Curriculum Guide, Graphics and Industrial Communications 1 and 2.

COMPUTER ORIENTED PROGRAMS
VT 010 458 32
Computer Games Management, Instructor's Manual.

COMPUTER SCIENCE
VT 010 709 81
Understanding Computers.

CONSERVATION EDUCATION
VT 000 949 4
Agricultural Production and Management.
VT 010 372 9
Directory of Conservation Education in Minnesota.

CONSUMER EDUCATION
VT 000 420 80
Teaching Consumer Education to Special Students on the Secondary Level.
VT 002 808 44
Dollars Demand Decisions.

QUARTERLY AUTHOR INDEX

PERSONAL AND INSTITUTIONAL AUTHOR INDEX

PAGE

AGRICULTURAL EDUCATION TEACHING MATERIALS CENTER, COLLEGE STATION, TEX.
VT 008 503 7
Agricultural Chemicals Employee.

AGRICULTURAL RESEARCH SERVICE (DOA), WASHINGTON, D.C. CONSUMER AND FOOD ECONOMICS RESEARCH DIV.
VT 011 294 46
A Guide to Budgeting for the Family.

AGRICULTURAL STABILIZATION AND CONSERVATION SERVICE (DOA), WASHINGTON, D.C.
VT 010 745 14
ASCS and You, A Teacher's Manual.
VT 010 746 15
ASCS and You, A Student Resource Unit.

ALABAMA STATE DEPT. OF EDUCATION, MONTGOMERY. AGRICULTURAL EDUCATION SERVICE
VT 000 378 2
A Specialized Course Outline for Advanced Vocational Agriculture Students in Forest Products Production Sales and Services.
VT 000 379 2
Forestry Production Sales and Services (A Student Study Guide.
VT 000 380 2
A Specialized Course Outline for Advanced Vocational Agriculture Students in Agricultural Supply, Sales and Services.
VT 000 381 3
Agriculture Supplies, Sales and Services (A Student Study Guide). Agribusiness Education Series.
VT 000 382 3
Ornamental Horticulture for Vocational Agriculture in Alabama.

ALABAMA UNIV., UNIVERSITY. DEPT. OF TRADE AND INDUSTRIAL EDUCATION
VT 010 582 74
Small Gasoline Engines; Study Guide and Job Tests.
VT 010 583 74
Building Maintenance, A Study Guide.
VT 010 590 74
Air Conditioning and Refrigeration.
VT 010 601 75
Volkswagen Mechanics, Study Guide and Job Tests.
VT 010 620 75
Everyday Arithmetic for Printers.
VT 010 621 75
Offset Printing, A Study Guide.
VT 010 627 76
Plumbing, A Study Guide.
VT 010 629 76
Weaving and Design Technology.

PAGE

VT 010 682 77
Textile Technician, A Study Guide.
VT 010 690 77
Automobile Body and Fender Repair and Refinishing, Job Tests.

ALBERTA DEPT. OF EDUCATION, EDMONTON
VT 010 439 53
Junior High School Curriculum Guide for Industrial Arts.

ALLEN, DELL K.
VT 002 608 56
Laboratory Manual for Experimental Metal-Cutting and Machine-Tool Evaluation.

ALLEY, WALTER
VT 010 263 70
How to Do the Complete Disc and Drum Brake Job.

AMERICAN ASSOCIATION FOR AGRICULTURAL ENGINEERING AND VOCATIONAL AGRICULTURE, ATHENS, GA.
VT 010 542 14
Tractor Fuels and Lubricants, Selecting and Storing.

AMERICAN DENTAL ASSISTANTS ASSOCIATION, CHICAGO, ILL. EDUCATION COMMITTEE
VT 011 220 42
Oral Histology.

AMERICAN SOCIETY OF RADIOLOGIC TECHNOLOGISTS, FOND DU LAC, WIS.
VT 007 363 50
Teaching Aids in Electricity.

AMMCO TOOLS INC., NORTH CHICAGO, ILL.
VT 010 268 71
Brake Service Manual.
VT 010 269 71
Disc Brake Manual.

AMMCO TOOLS, INC., NORTH CHICAGO, ILL.
VT 010 263 70
How to Do the Complete Disc and Drum Brake Job.
VT 010 267 70
Wheel Alignment Manual.

ARGYLE PUBLISHING CORP., NEW YORK, N.Y.
VT 011 042 41
Selected Medical Terminology.

ARIZONA UNIV., TUCSON. DEPT. OF AGRICULTURAL EDUCATION
VT 010 259 8
Course Guide for Basic Agricultural Mechanics.

PERSONAL AND INSTITUTIONAL AUTHOR INDEX

PAGE

ORANGE COUNTY BOARD OF EDUCATION, CALIF. EDUCATIONAL SERVICES DIV.
VT 000 420 80
Teaching Consumer Education to Special Students on the Secondary Level.

OREGON STATE BOARD OF EDUCATION, SALEM. DIV. OF COMMUNITY COLLEGES AND CAREER EDUCATION
VT 011 175 ED 040 285 41
Curriculum Guide for Health Occupations.

PEALE, BURMA ROSE, AND OTHERS
VT 000 420 80
Teaching Consumer Education to Special Students on the Secondary Level.

PENNSYLVANIA STATE UNIV., UNIVERSITY PARK. DEPT. OF AGRICULTURAL EDUCATION
VT 010 745 14
ASCS and You, A Teacher's Manual.
VT 010 746 15
ASCS and You, A Student Resource Unit.

PENNSYLVANIA YOUNG FARMERS ASSOCIATION
VT 010 362 9
Legal Principles for Farm Managers.

PENROD, L.O.
VT 010 620 75
Everyday Arithmetic for Printers.

PHIPPS, D. BART
VT 009 973 63
Basic Refrigeration for Apprentice Training in the Plumbing and Pipe Fitting Industry.
VT 009 976 64
Sketching and Plan Reading for Apprentice Training in the Plumbing and Pipe Fitting Industry.
VT 009 996 64
Fundamentals of Plumbing Installation for Apprentice Training in the Plumbing and Pipe Fitting Industry.
VT 010 012 65
Industrial Blueprints for Apprentice Training in the Plumbing and Pipe Fitting Industry.
VT 010 013 65
Basic Instruction for Apprentice Training in the Plumbing and Pipe Fitting Industry.
VT 010 014 65
Drawing and Plan Reading for Apprentice Training in the Plumbing and Pipe Fitting Industry.
VT 010 015 66
Rigging and Safety for Apprentice Training in the Plumbing and Pipe Fitting Industry.
VT 010 026 67
Principles of Heating for Apprentice Training in the Plumbing and Pipe Fitting Industry.

PORTLAND STATE COLL., ORE.
VT 009 359 40
Nursing Home and Extended Care Administration.

POYNER, CLINTON B., SR., COMP.
VT 008 422 61
Food Trades. Curriculum Guides in Trade and Industrial Education.

PROGRAMS FOR ACHIEVEMENT IN READING, INC., PROVIDENCE, R.I.
VT 011 142 25
Typewriting for Business; Typewriting by a Whole (Gestalt) Method. Teacher's Manual and Key.
VT 011 143 25
Timed Writing, No. 1-7.
VT 011 145 25
Typewriting for Business; Typewriting by a Whole (Gestalt) Method.
VT 011 151 26
Phonics Review for Shorthand.
VT 011 152 26
Phonics Review for Shorthand. Teacher's Edition.

PRUDHOMME, RHETTA W., COMP.
VT 008 501 62
Household Operations. Curriculum Guides in Trade and Industrial Education.

PULASKI, RICHARD E.
VT 009 975 64
Theory and Practice of Natural Gas Installation for Apprentice Training in the Plumbing and Pipe Fitting Industry.
VT 010 024 66
Basic Instruction: Apprentice Training in the Plumbing and Pipe Fitting Industry.
VT 010 025 66
Basic Instruction Apprentice Training in the Plumbing and Pipe Fitting Industry.

RAWLS, BYRON F.
VT 000 378 2
A Specialized Course Outline for Advanced Vocational Agriculture Students in Forest Products Production Sales and Services.
VT 000 379 2
Forestry Production Sales and Services (A Student Study Guide.

QUARTERLY SUPPORTING SERVICES INDEX

PAGE

AGRICULTURAL EDUCATION

VT 004 032 4
Supervised Occupational Experience Program in Vocational Agriculture for Arkansas, (Type C Program).

VT 004 591 282
A Pilot Program in New Teacher Preparation and In-Service Upgrading of Teachers and Supervisors.

VT 005 125 176
Evaluation of 4-H Veterinary Science Project in Three Counties of New York State.

VT 007 404 14
Guidelines for Program Development; High School Vocational Horticulture.

VT 007 476 ED 024 825 15
NEED FOR A POST-HIGH SCHOOL VOCATIONAL TRAINING PROGRAM IN HORTICULTURE.

VT 007 800 18
Adult Education in Agriculture.

VT 008 280 ED 029 134 97
DEVELOPING OCCUPATIONAL EXPERIENCE PROGRAMS IN AGRICULTURAL EDUCATION. BULL. 72H-5.

VT 008 368 131
Opportunities in Agricultural Education.

VT 008 427 132
Educational Needs and Employment Opportunities in Non-Farm Agriculture.

VT 008 458 ED 029 149 28
ONE-YEAR AND TWO-YEAR POST HIGH SCHOOL INSTITUTIONS WHICH OFFER PROGRAMS OF INSTRUCTION IN AGRICULTURE, 1968-69 DIRECTORY.

VT 008 561 133
South Dakota Agricultural Off-Farm Occupational Opportunities andTraining Needs.

VT 008 738 ED 030 738 35
A POLICY BULLETIN AND RECOMMENDATIONS FOR VOCATIONAL AGRICULTURE DEPARTMENTS IN WYOMING.

VT 008 800 192
A Study to Determine the Physical Facilities Needed for Vocational Agriculture.

VT 008 804 ED 030 747 146
REQUIREMENTS AND OPPORTUNITIES FOR ENTRY WORKERS IN THE OCCUPATION OF PRODUCING AND MARKETING ORNAMENTAL NURSERY AND GREENHOUSE SPECIALISTS.

VT 008 929 ED 030 761 197
HISTORICAL DEVELOPMENT OF AGRICULTURAL EDUCATION IN THE UNITED STATES PRIOR TO 1917. FINAL REPORT.

VT 008 957 237
The Commission on Education in Agriculture and Natural Resources.

PAGE

VT 008 968 ED 030 764 287
THE IDENTIFICATION OF OPINION LEADERS AMONG TEACHERS OF VOCATIONAL AGRICULTURE. FINAL REPORT. RESEARCH 40.

VT 008 996 ED 031 585 153
FACTORS AFFECTING THE ESTABLISHMENT OF YOUNG FARM OPERATORS IN IOWA AND IMPLICATIONS FOR AGRICULTURAL EDUCATION.

VT 009 782 290
Annual Report, Department of Agricultural Education, 1968-1969.

VT 009 920 ED 035 729 193
A TEACHING EXPERIMENT IN THE USE OF GREENHOUSE FACILITIES IN VOCATIONAL AGRICULTURE.

VT 009 921 ED 035 730 300
AN EXPERIMENT IN TEACHING AGRICULTURAL OCCUPATIONS INFORMATION TO HIGH SCHOOL STUDENTS.

VT 009 990 ED 035 886 163
EDUCATION FOR OFF-FARM AGRICULTURAL OCCUPATIONS IN NEVADA.

VT 010 166 ED 036 001 73
AGRICULTURAL OCCUPATIONS PROGRAM DEVELOPMENT IN AREA VOCATIONAL SCHOOLS. FINAL REPORT.

VT 010 205 ED 035 761 74
AN EXPERIMENTAL GUIDE FOR PERSONNEL TRAINING REQUIREMENTS OF TECHNICIANS IN FUTURE FOOD IRRADIATION TECHNOLOGY INDUSTRIES. FINAL REPORT.

VT 010 280 ED 035 769 101
ATTITUDES OF SELECTED GROUPS CONCERNING THE ROLE OF VOCATIONAL-TECHNICAL EDUCATION PROGRAMS FOR OCCUPATIONS IN AGRICULTURAL RESOURCES.

VT 010 297 292
Farm Power Mechanics Competencies of Minnesota Vocational Agriculture Instructors.

VT 010 299 76
Vocational Horticulture Specialty Programs for Secondary Students.

VT 010 311 ED 035 773 102
AN EXPERIMENTAL EVALUATION OF APPROACHES TO PREPARING HIGH SCHOOL STUDENTS FOR AGRICULTURAL OCCUPATIONS OTHER THAN FARMING AND (PRINCIPLES VERSUS TRADITIONAL APPROACH TO TEACHING VOCATIONAL AGRICULTURE). FINAL REPORT.

VT 010 329 253
Research Studies in Progress in Agricultural Education in the North Atlantic Region, 1969-70.

VT 010 383 ED 038 504 77
DUTCH AGRICULTURAL EDUCATION.

VT 010 398 ED 036 647 186
THE RELEVANCE OF SECONDARY OCCUPATIONAL TRAINING IN AGRICULTURE TO OCCUPATIONAL PATTERNS AND IMAGES. FINAL REPORT.

VOCATIONAL AND SUPPORTING SERVICES INDEX

PAGE

VT 010 283 ED 036 633 223
Comprehensive Statewide Planning for Vocational Rehabilitation Services. Final Report, Volume II.

VT 010 284 ED 036 634 223
Comprehensive Statewide Planning for Vocational Rehabilitation Services. Final Report, Volume I.

VT 010 285 ED 037 553 185
Demographic Techniques for Manpower Planning in Developing Countries.

VT 010 286 ED 035 771 240
EDUCATION, HUMAN RESOURCES, AND DEVELOPMENT IN LATIN AMERICA.

VT 010 349 ED 039 323 224
Hard-Core Unemployment: A Selected, Annotated Bibliography.

VT 010 368 67
Employment and Manpower Problems in the Cities: Implications of the Report of the National Advisory Commission on Civil Disorders.

VT 010 495 ED 037 562 78
GOVERNOR'S MANPOWER CONFERENCE REPORT. 1969.

VT 010 541 ED 038 508 79
Summaries of Manpower Surveys and Reports for Developing Countries, 1958-68.

VT 010 585 ED 038 51[illegible] 225
Poor People at Work, An Annotated Bibliography on Semi-Professionals in Education, Health, and Welfare Services.

VT 010 608 80
Related Instruction for Apprentices.

VT 010 645 168
A Survey of the Manpower Needs in the Commercial Nursery Industry.

VT 010 658 168
Demand for Engineers and Technicians 1968.

VT 010 678 ED 039 329 225
Private Industry and the Disadvantaged Worker.

VT 010 708 ED 039 331 169
THE PRE-RETIREMENT YEARS: A LONGITUDINAL STUDY OF LABOR MARKET EXPERIENCE OF MEN. VOLUME TWO.

VT 010 718 169
Job Descriptions: How to Write and Use Them.

VT 010 731 ED 037 579 169
Health Manpower Source Book. Manpower Supply and Educational Statistics for Selected Health Occupations: 1968.

VT 010 753 ED 038 525[illegible] 83
A STUDY OF MANPOWER DEVELOPMENT AND TRAINING ACT PROGRAMS IN NEW YORK STATE.

VT 010 800 ED 038 529 85
STRATEGIES FOR MEETING CORRECTIONAL TRAINING AND MANPOWER NEEDS, FOUR DEVELOPMENTAL PROJECTS.

PAGE

VT 010 806 ED 039 334 189
Cost-Benefit Analysis of Manpower Policies. Proceedings of a North American Conference.

VT 010 869 ED 039 338 170
JOBS AND CIVIL RIGHTS; THE ROLE OF THE FEDERAL GOVERNMENT IN PROMOTING EQUAL OPPORTUNITY IN EMPLOYMENT AND TRAINING. CLEARINGHOUSE PUBLICATION NO. 16.

VT 010 875 ED 039 340 170
GUIDE TO LOCAL OCCUPATIONAL INFORMATION.

VT 010 902 ED 039 342 227
EVALUATION OF THE MISSISSIPPI-IOWA FARM LABOR MOBILITY PROJECT.

OCCUPATIONAL PSYCHOLOGY

VT 007 108 176
Perceptual-Psychomotor Tests in Aircrew Selection.

VT 007 927 [illegible]3
Nursing Attitudes and Turnover.

VT 007 988 ED 027 43[illegible] 125
VOCATIONAL INTERESTS OF NON-PROFESSIONAL WOMEN. FINAL REPORT.

VT 008 002 266
Relationship of Vocational Satisfaction to the Correspondence of Job Reinforcement and Vocational Needs.

VT 008 512 ED 030 718 214
DEPENDENCY AND ITS IMPLICATIONS FOR REHABILITATION. NORTHEASTERN STUDIES IN VOCATIONAL REHABILITATION. MONOGRAPH NO. 1.

VT 008 798 ED 030 746 38
RESEARCH PROJECTS 1968.

OCCUPATIONAL SOCIOLOGY

VT 004 741 263
A Reconceptualization of Vocational Guidance.

VT 005 742 283
A Teacher's Role with the Disadvantaged.

VT 006 952 MP 000 610 264
Retraining Under the Manpower Development Training Act.

VT 007 663 ED 027 397 266
SOCIOLOGICAL STUDIES OF OCCUPATIONS; ABSTRACTS.

VT 008 003 125
Supervisor Perceptions of Occupational Environments and Roe's Classification of Occupations.

VT 008 171 ED 029 123 129
IN SEARCH OF OPPORTUNITY. A STUDY OF POST HIGH SCHOOL MIGRATION IN MINNESOTA. TECHNICAL BULLETIN 247.

VT 008 578 134
They Carry the Burden Alone: The Socio-Economic Living Pattern of Oregon Women with Dependents.

VT 008 580 ED 029 981 135
THE MODEL NEIGHBORHOOD JOBSEEKER. A SURVEY OF HAWAII STATE EMPLOYMENT SERVICE JOB APPLICANTS.

QUARTERLY DOCUMENT NUMBER INDEX

DOCUMENT NUMBER INDEX

	PAGE
VT 000 013 Let's Analyze and Sell.	28
VT 000 239 Industrial Arts Supplement.	48
VT 000 270 Manual for Instructors of Vocational Nursing, Pre-Clinical Period, 16-20 Weeks.	40
VT 000 282 A Training Program for Vocational Agriculture in Wildlife and Recreation.	2
VT 000 330 Instructor's Source Book, Vocational Nurse Training Program.	40
VT 000 378 A Specialized Course Outline for Advanced Vocational Agriculture Students in Forest Products Production Sales and Services.	2
VT 000 379 Forestry Production Sales and Services (A Student Study Guide.	2
VT 000 380 A Specialized Course Outline for Advanced Vocational Agriculture Students in Agricultural Supply, Sales and Services.	2
VT 000 381 Agriculture Supplies, Sales and Services (A Student Study Guide). Agribusiness Education Series.	3
VT 000 382 Ornamental Horticulture for Vocational Agriculture in Alabama.	3
VT 000 420 Teaching Consumer Education to Special Students on the Secondary Level.	80
VT 000 946 Agriculture II: A Curriculum Unit Outline for Vocational Agriculture.	4
VT 000 949 Agricultural Production and Management.	4
VT 000 989 Agricultural Instruction, Units I-VIII.	4
VT 001 093 Course of Study for Ornamental Horticulture Courses.	4
VT 001 238 Source Units for the Dairy Enterprise.	5
VT 001 323 Test Construction in Distributive Education.	28
VT 001 346 Agricultural Chemicals.	5
VT 001 359 Industrial Arts.	48
VT 001 571 Vocational Agriculture Course of Study Guide.	5
VT 001 873 Education for Business.	20
VT 001 881 Bookkeeping I, Syllabus.	20
VT 001 998 Home Economics Humanities Media Guide.	44
VT 002 053 Water and Sewerage, A Course of Study.	56
VT 002 112 A Guide for Teaching Power Mechanics in Industrial Arts in Indiana.	48
VT 002 345 Industrial Arts Safety Instruction.	49
VT 002 568 Marketing II, Teachers Manual.	28
VT 002 588 Cooperative Distributive Education; Teacher's Guide to Student Activities.	28
VT 002 608 Laboratory Manual for Experimental Metal-Cutting and Machine-Tool Evaluation.	56
VT 002 645 Civil Engineering Technology, Field Book.	56
VT 002 650 Guide Lines for the Junior High School Industrial Arts Program.	49
VT 002 656 Civil Engineering Technology.	56

DOCUMENT NUMBER INDEX

	PAGE
VT 010 478 Related Guide..Job Employment, Distributive Education.	33
VT 010 480 Source Unit in Animal Breeding.	11
VT 010 484 Source Unit in Feeds and Feeding	12
VT 010 505 Modern Metalworking Technology.	53
VT 010 527 Tree Identification: Western Pines.	12
VT 010 528 Careers in Animal Industry.	12
VT 010 529 Careers in Government Service.	81
VT 010 530 Careers in Natural Resources Management.	81
VT 010 531 Careers in Ornamental Horticulture.	13
VT 010 532 Careers in Farm Supplies.	13
VT 010 533 Careers in Farm Services.	13
VT 010 534 Careers in Farm Machinery Sales and Service.	13
VT 010 535 Careers in Crops.	14
VT 010 542 Tractor Fuels and Lubricants, Selecting and Storing.	14
VT 010 543 What a Fireman Should Know About Sprinkler Systems.	72
VT 010 544 Fire Department Driver Training Manual of Instruction.	72
VT 010 545 Manual of Salvage Operations.	72
VT 010 546 Recommendation 19: The Training of Power Station Operators Coal, Oil and Gas Fired Stations.	73
VT 010 570 Franchise Merchandising. A Suggested Adult Course Outline.	34
VT 010 571 California Fire Training Program; Book 2 of Block III, Use of Fire Apparatus and Equipment.	73
VT 010 572 California Fire Training Program; Book 1 of Block III, Use of Fire Apparatus and Equipment.	73
VT 010 582 Small Gasoline Engines; Study Guide and Job Tests.	74
VT 010 583 Building Maintenance, A Study Guide.	74
VT 010 590 Air Conditioning and Refrigeration.	74
VT 010 601 Volkswagen Mechanics, Study Guide and Job Tests.	75
VT 010 611 Business and Office Education: Before and After in Indiana.	22
VT 010 615 Vocational Office Education in Florida High Schools.	22
VT 010 619 Variety Store Merchandising.	34
VT 010 620 Everyday Arithmetic for Printers.	75
VT 010 621 Offset Printing, A Study Guide.	75
VT 010 627 Plumbing, A Study Guide.	76
VT 010 629 Weaving and Design Technology.	76
VT 010 638 Servicing Carburetor Air Cleaners.	76
VT 010 682 Textile Technician, A Study Guide.	77
VT 010 690 Automobile Body and Fender Repair and Refinishing, Job Tests.	77
VT 010 709 Understanding Computers.	81
VT 010 726 Nitrates in Water Supplies, Field Crops, and Ruminant Nutrition.	14
VT 010 734 Introduction to Programming PDP-8 Family Computers.	22

CONVERSION OF DOCUMENT NUMBER

INDEX

CONVERSION OF DOCUMENT NUMBER INDEX[1]

VT NUMBERS CONVERTED TO ED NUMBERS OR MP NUMBERS[2]

VT 002 431	ED 035 715	VT 008 273	ED 029 964	VT 008 804	ED 030 747
VT 002 508	ED 029 943	VT 008 280	ED 029 134	VT 008 815	ED 039 316
VT 004 667	ED 038 488	VT 008 303	ED 032 395	VT 008 830	ED 030 750
VT 006 275	ED 025 609	VT 008 429	ED 029 147	VT 008 831	ED 030 751
VT 006 277	ED 025 611	VT 008 433	ED 029 148	VT 008 832	ED 030 752
VT 006 285	ED 035 788	VT 008 438	ED 029 970	VT 008 833	ED 030 753
VT 006 952	MP 000 610	VT 008 444	ED 030 716	VT 008 844	ED 031 574
VT 007 147	ED 025 632	VT 008 451	ED 028 306	VT 008 850	ED 031 577
VT 007 154	ED 023 896	VT 008 458	ED 029 149	VT 008 854	ED 032 407
VT 007 211	ED 032 390	VT 008 511	ED 030 717	VT 008 860	ED 030 756
VT 007 315	ED 027 388	VT 008 512	ED 030 718	VT 008 863	ED 031 578
VT 007 476	ED 024 825	VT 008 513	ED 029 976	VT 008 868	ED 036 618
VT 007 547	ED 026 517	VT 008 538	ED 038 489	VT 008 874	ED 030 757
VT 007 555	ED 025 654	VT 008 539	ED 038 490	VT 008 899	ED 032 410
VT 007 560	ED 033 188	VT 008 544	ED 029 977	VT 008 902	ED 030 759
VT 007 582	ED 025 658	VT 008 547	ED 030 722	VT 008 906	ED 031 579
VT 007 662	ED 037 522	VT 008 554	ED 029 978	VT 008 921	ED 030 760
VT 007 663	ED 027 397	VT 008 576	ED 031 566	VT 008 926	ED 031 580
VT 007 676	ED 025 672	VT 008 579	ED 030 723	VT 008 929	ED 030 761
VT 007 677	ED 025 673	VT 008 580	ED 029 981	VT 008 930	ED 033 199
VT 007 747	ED 034 019	VT 008 592	ED 030 724	VT 008 947	ED 032 073
VT 007 776	ED 026 531	VT 008 593	ED 030 725	VT 008 951	ED 032 413
VT 007 790	ED 025 679	VT 008 596	ED 032 397	VT 008 952	ED 030 762
VT 007 797	ED 029 103	VT 008 600	ED 029 989	VT 008 953	ED 030 763
VT 007 805	ED 039 312	VT 008 601	ED 029 990	VT 008 968	ED 030 764
VT 007 831	ED 029 106	VT 008 607	ED 030 726	VT 008 970	ED 030 765
VT 007 881	ED 029 956	VT 008 609	ED 031 568	VT 008 971	ED 032 414
VT 007 883	ED 026 541	VT 008 631	ED 029 993	VT 008 974	ED 030 766
VT 007 887	ED 026 542	VT 008 632	ED 029 994	VT 008 976	ED 030 767
VT 007 920	ED 036 613	VT 008 635	ED 033 196	VT 008 980	ED 031 583
VT 007 931	ED 027 425	VT 008 647	ED 030 729	VT 008 982	ED 030 768
VT 007 971	ED 039 313	VT 008 693	ED 030 731	VT 008 996	ED 031 585
VT 007 988	ED 027 433	VT 008 707	ED 031 569	VT 009 016	ED 031 588
VT 008 008	ED 029 112	VT 008 718	ED 033 197	VT 009 017	ED 031 589
VT 008 051	ED 029 116	VT 008 719	ED 031 570	VT 009 019	ED 032 415
VT 008 082	ED 028 288	VT 008 727	ED 032 398	VT 009 057	ED 031 593
VT 008 091	ED 032 392	VT 008 728	ED 032 399	VT 009 075	ED 033 200
VT 008 127	ED 028 292	VT 008 729	ED 032 400	VT 009 082	ED 033 201
VT 008 129	ED 029 121	VT 008 730	ED 032 401	VT 009 086	ED 031 595
VT 008 138	ED 029 122	VT 008 735	ED 030 736	VT 009 087	ED 031 596
VT 008 146	ED 032 393	VT 008 736	ED 030 737	VT 009 126	ED 033 202
VT 008 150	ED 028 294	VT 008 738	ED 030 738	VT 009 154	ED 032 419
VT 008 171	ED 029 123	VT 008 750	ED 036 616	VT 009 158	ED 031 600
VT 008 173	ED 028 297	VT 008 753	ED 032 402	VT 009 159	ED 033 204
VT 008 177	ED 032 394	VT 008 769	ED 032 403	VT 009 185	ED 031 601
VT 008 182	ED 029 124	VT 008 770	ED 030 743	VT 009 188	ED 031 602
VT 008 183	ED 028 298	VT 008 776	ED 032 405	VT 009 190	ED 032 420
VT 008 185	ED 031 562	VT 008 777	ED 032 406	VT 009 192	ED 031 603
VT 008 198	ED 029 126	VT 008 787	ED 031 572	VT 009 196	ED 033 209
VT 008 222	ED 029 130	VT 008 798	ED 030 746	VT 009 198	ED 032 421

[1] This index is cumulated annually beginning with the Fall issue of ARM.

[2] ...cument with an MP number is available from EDRS by using the ...riate MP number.

ANNUAL SUBJECT INDEX

SUBJECT AND IDENTIFIER INDEX

ABRAHAM JACOBS MEMORIAL SYMPOSIUM
66: 5436

ABSENTEEISM
66: 3752

ABSTRACTS
66: 7309
67: 4817
68: 7262, 7466, 7618, 7619, 7620, 7621
8602
69: 8588
10149, 10295, 10322, 10328

AC GENERATORS
63: 8806

ACADEMIC ABILITY
68: 6571

ACADEMIC ACHIEVEMENT
60: 4618
66: 3314
4576
67: 3758
68: 8151
69: 8373

ACADEMIC APTITUDE
68: 6600
69: 8277

ACADEMIC ASPIRATION
67: 7266, 7530
68: 6571

ACADEMIC EDUCATION
67: 6814
68: 7660

ACCELERATED COURSES
63: 2539

ACCELERATED PROGRAMS
65: 4508

ACCELERATED VOCATIONAL TRAINING
65: 4508

ACCIDENT PREVENTION
nd: 9942
63: 150
64: 1678
66: 10041
68: 8071, 8724

ACCOUNTANTS
68: 9363, 9364

ACCOUNTING
nd: 8060
65: 9229
68: 5955

ACCREDITATION (INSTITUTIONS)
64: 1532, 1900
65: 4652
5997
6027
66: 2638
6254
67: 4746
5999
6000
7821
68: 7497, 7819
9231, 9232, 9237

ACHIEVEMENT
68: 6862
7504

EVEMENT GAINS
6671

ACHIEVEMENT RATING
69: 8648

ACHIEVEMENT TESTS
63: 5237
8276
64: 5560
68: 7228
8505
69: 9610

ACTION PROGRAMS (COMMUNITY)
67: 6786

ACTIVITIES
67: 4996

ADHESIVES
68: 7858

ADMINISTRATION
67: 4792

ADMINISTRATIVE CHANGE
68: 5870

ADMINISTRATIVE ORGANIZATION
nd: 5483
66: 2553
6691
67: 6573
69: 8391

ADMINISTRATIVE PERSONNEL
66: 877
68: 7372, 7679
69: 7837, 7970

ADMINISTRATIVE POLICY
nd: 5489
63: 2028, 2383
64: 1521
6462
66: 6348
67: 3615
5316
7084
68: 8659

ADMINISTRATIVE PROBLEMS
65: 6616
68: 7079, 7900
69: 9839
10456

ADMINISTRATOR ATTITUDES
nd: 8739
65: 4575
67: 7067
68: 6677, 6769
7322, 7566, 7693

ADMINISTRATOR BACKGROUND
69: 9839

ADMINISTRATOR CHARACTERISTICS
69: 9839

ADMINISTRATOR GUIDES
nd: 1463
2641
63: 1702
4121
64: 2496
5615
65: 1606
2132, 2597
66: 1319, 1509, 1552, 1673
2504, 2592, 2593
5283
7326
67: 6657
68: 7787
8050, 8062
69: 8802

ADMINISTRATOR GUIDES (CONTINUED)
69: 9894, 9899
10511

ADMINISTRATOR RESPONSIBILITY
nd: 1463
68: 5874

ADMINISTRATOR ROLE
nd: 1463
67: 2727
68: 5874

ADMISSION (SCHOOL)
68: 7365

ADMISSION CRITERIA
65: 2541
66: 1897
4019
68: 6686
7582

ADOLESCENTS
67: 8522

ADOPTION (IDEAS)
67: 9843
68: 6014

ADULT BASIC EDUCATION
64: 1555
66: 8653

ADULT CHARACTERISTICS
68: 6506

ADULT EDUCATION
nd: 7176
65: 1695
6426
68: 6720
7741
69: 9491
10558

ADULT EDUCATION PROGRAMS
62: 10102
66: 10103, 10104, 10105, 10108, 10121, 10122, 10123
67: 10106, 10107, 10120
68: 7059

ADULT FARMER EDUCATION
nd: 544
65: 2386
8061
9252
66: 4585
6823
68: 5855
8646, 8656
69: 10098, 10426, 10506

ADULT LEARNING
64: 4535

ADULT STUDENTS
68: 7059

ADULT VOCATIONAL EDUCATION
nd: 5579
8418, 8573, 8910, 8911
63: 831
2013
8482
64: 2909
3608
9047, 9090
65: 3901
4508
8959, 8960, 8961, 8985, 8986
9001, 9002, 9003, 9023, 9024, 9025, 9026, 9027,

AIR CONDITIONING (CONTINUED)
68: 8342

AIRCRAFT PILOTS
67: 6804
69: 9490, 9582, 9588

ALGEBRA
69: 9819

ALLIED HEALTH PROFESSIONS PROJECTS
68: 7625

AMA ANNUAL CONFERENCE ON MEDICAL EDUCATION, CHICAGO, ILLINOIS, FEBRUARY 9, 1969
69: 8641

AMERICAN ASSOCIATION OF DENTAL SCHOOLS
68: 7274

AMERICAN ASSOCIATION OF JUNIOR COLLEGES, EAST LANSING, MICHIGAN, FEBRUARY 21, 1969
69: 8556

AMERICAN ASSOCIATION OF TEACHER EDUCATORS IN AGRICULTURE
68: 7696

AMERICAN DENTAL ASSOCIATION
67: 5316

AMERICAN HISTORY
66: 1914

AMERICAN INDIANS
68: 6999
8791

AMERICAN INDUSTRIAL ARTS ASSOCIATION
69: 9853

AMERICAN INDUSTRIAL ARTS STUDENT ASSOCIATION
67: 1954

AMERICAN INDUSTRIAL TECHNICAL EDUCATION CONFERENCE, COLUMBUS, OHIO, JUNE 20-21, 1967
67: 6123

AMERICAN INDUSTRY PROJECT
65: 2861

AMERICAN SOCIETY FOR ENGINEERING EDUCATION, UNIVERSITY OF CALIFORNIA, LOS ANGELES, JUNE 17-20, 1968
68: 7247

AMERICAN SOCIETY TOOL AND MANUFACTURING ENGINEERS
68: 8762

AMERICAN STATISTICAL ASSOCIATION AND ALLIED SOCIETIES
67: 6840

AMERICAN VOCATIONAL ASSOCIATION
68: 7380

AMERICAN VOCATIONAL ASSOCIATION CONVENTION, DALLAS, TEXAS, DECEMBER 9-13, 1968
68: 8586

AMERICAN VOCATIONAL ASSOCIATION CONVENTION, DALLAS, TEXAS, 1968
68: 8643

AMPLIFIERS
66: 10104

ANATOMY
64: 9326

ANATOMY (CONTINUED)
65: 9308, 9678

ANIMAL CARETAKERS
nd: 9320, 9327

ANIMAL SCIENCE
nd: 9325
10424
63: 1400
65: 57
66: 9775
68: 8764, 8766, 8767, 8771, 8822, 8933, 8934, 8935, 8936, 8958
69: 7993
8765, 8949
9629, 9871, 9872

ANNOTATED BIBLIOGRAPHIES
62: 6412
64: 6325
65: 2555
3782
66: 2721
4196, 4684, 4978
5225
5465
7699
67: 2389
3779
6331, 6810
8083, 8890
68: 5631
7118, 7228, 7234, 7239, 7262, 7420, 7466, 7618, 7619, 7620, 7621, 7903 8117, 8577, 8888
69: 7889
8568, 8591
9006, 9007, 9055, 9352, 9404, 9432, 9563, 9622, 9627, 9857, 9936
10077, 10295, 10322, 10328, 10364, 10430, 10464, 10558
70: 10509, 10829
11173, 11200

ANNUAL AMERICAN VOCATIONAL ASSOCIATION CONVENTION (63RD, BOSTON, MASSACHUSETTS, DECEMBER 6-9, 1969
69: 10450

ANNUAL AMERICAN VOCATIONAL ASSOCIATION CONVENTION (63RD, BOSTON, MASSACHUSETTS, DECEMBER 6-9, 1969)
65: 10443
69: 10110, 10111, 10174, 10176, 10298, 10302, 10305, 10307, 10392, 10393, 10394, 10413, 10414, 10415, 10416, 10431, 10432, 10433, 10434, 10435, 10440, 10441, 10444, 10445, 10446, 10447, 10448, 10449, 10451, 10452, 10453, 10454, 10456

ANNUAL AMERICAN VOCATIONAL ASSOCIATION CONVENTION (63RD, BOSTON, MASSACHUSETTS, DECEMBER 6-9, 1969).
69: 10455

ANNUAL CONFERENCE ON INDUSTRIAL EDUCATION, ANAHEIM, CALIFORNIA, MARCH 4-6, 1964
64: 4611

ANNUAL CONFERENCE ON INDUSTRIAL EDUCATION, SACRAMENTO, CALIFORNIA, MARCH 14-16, 1967
67: 2214

ANNUAL CONVENTION OF THE NATIONAL ASSOCIATION OF PSYCHIATRIC TECHNOLOGY, FRESNO, CALIFORNIA, OCTOBER, 1967
nd: 10316

ANNUAL REPORTS
65: 2109
66: 1367, 1621
67: 587
2410, 2428
5253, 5316
68: 5462
7622, 7741
8054
69: 9991

ANNUAL RESEARCH CONFERENCE
67: 6624

ANSWER KEYS
68: 7958

ANXIETY
68: 6669

APPRENTICESHIPS
nd: 8810, 8810, 8811, 8812
53: 9875
62: 717
63: 4800
64: 774
3524, 3808
65: 4408, 4600, 4798
66: 1367, 1552
3245, 3280, 3554
4799, 4801, 4876
8415, 8651
67: 3595
4712, 4803
7084
8841
9374
10222, 10224, 10225
68: 7414
8374, 8569, 8633, 8676, 8838
9022, 9366, 9367, 9368, 9371, 9372, 9373, 9376, 9378, 9379
10155, 10220, 10221, 10226, 10227, 10234, 10236
69: 9377
10214

APTITUDE
67: 2362

APTITUDE TESTS
67: 5978
69: 7817
8629

APTITUDES
67: 7044

ARCHITECTURAL EDUCATION
69: 10769

AREA VOCATIONAL SCHOOLS
nd: 8244
63: 1702
5178
64: 4675, 4679
65: 2359
4960
66: 1509, 1718, 1827
4666
67: 1992
4805
68: 5860
7482
8015, 8390, 8392
69: 8389, 8391, 8629
9169

ARIZONA
64: 3608

ANNUAL AUTHOR INDEX

PERSONAL AND INSTITUTIONAL AUTHOR INDEX

ABILENE PUBLIC SCHOOLS, TEX.
65: 352

ABRAMS, HARVEY A.
68: 8775

ACHIEVEMENT MOTIVATION DEVELOPMENT PROJECT, CAMBRIDGE, MASS.
69: 8366

ADAMS, JERRY
nd: 8250

ADAMS, JON P.
69: 7998

ADAMS, WALTER, ED.
67: 2690

ADKINS, DOLORES, AND OTHERS
65: 107

ADVISORY COUNCIL ON VOCATIONAL EDUCATION, WASHINGTON, D.C.
68: 7877

AEROSPACE MEDICAL RESEARCH LABS., WRIGHT-PATTERSON AFB, OHIO
67: 7098

AEROSPACE MEDICAL RESEARCH LABS, WRIGHT PATTERSON AFB, OHIO.
67: 7099

AEROSPACE MEDICAL RESEARCH LABS, WRIGHT-PATTERSON AFB, OHIO
67: 7101

AGAN, R.J., AND OTHERS
63: 2213

AGAN, RAYMOND J.
68: 6931

AGENCY FOR INTERNATIONAL DEVELOPMENT, WASHINGTON, D.C. COMMUNICATIONS RESOURCES DIV.
62: 2733

AGENCY FOR INTERNATIONAL DEVELOPMENT, WASHINGTON, D.C. CRD-PUBLICATIONS AND TECHNICAL SERVICES BRANCH
63: 2664

AGENCY FOR INTERNATIONAL DEVELOPMENT, WASHINGTON, D.C. OFFICE OF LABOR AFFAIRS
66: 5478

AGRICULTURAL EDUCATION TEACHING MATERIALS CENTER, COLLEGE STATION, TEX.
68: 8252

AGRICULTURAL RESEARCH SERVICE (DOA), WASHINGTON, D.C.
65: 9943
67: 10404

AIKEN, MICHAEL
67: 5475

AIKIN, BRENTON R.
63: 3691

AINSWORTH, ROBERT G.
67: 6856

AIR-CONDITIONING AND REFRIGERATION INST., ARLINGTON,
68: 5853

AIR-CONDITIONING AND REFRIGERATION INST., ARLINGTON, VA. (CONTINUED)
68: 7607

AKAMINE, TOSHIO
68: 6919

AKRON PUBLIC SCHOOLS, OHIO
64: 445, 446, 447

ALABAMA RESEARCH COORDINATING UNIT FOR VOCATIONAL AND TECHNICAL EDUCATION, MONTGOMERY
69: 9791

ALABAMA STATE DEPT. OF EDUCATION, MONTGOMERY
nd: 176

ALABAMA STATE DEPT. OF EDUCATION, MONTGOMERY. AGRICULTURAL EDUCATION SERVICE
66: 367, 368, 372, 373, 374, 375, 376, 377, 383

ALABAMA STATE DEPT. OF EDUCATION, MONTGOMERY. TRADE AND INDUSTRIAL EDUCATION
67: 10589

ALABAMA STATE EMPLOYMENT SERVICE, MONTGOMERY
68: 7503

ALABAMA UNIV., UNIVERSITY DEPT. OF TRADE AND INDUSTRIAL EDUCATION
67: 10589

ALABAMA UNIV., UNIVERSITY. DEPT. OF TRADE AND INDUSTRIAL EDUCATION
68: 10581,10691
69: 10592,10687

ALAMEDA COUNTY HEALTH DEPT., OAKLAND, CALIF.
68: 7381

ALASKA OFFICE OF THE GOVERNOR, JUNEAU
65: 2511

ALASKA STATE DEPT. OF EDUCATION, JUNEAU. DIV. OF VOCATIONAL EDUCATION
66: 5283
8640

ALBORN A. AND M. COLL., LORMAN, MISS.
69: 8277

ALBRACHT, JAMES
68: 7980

ALCORN A AND M COLL., LORMAN, MISS. DIV. OF VOCATIONAL EDUCATION
65: 4558

ALLEN, DAVID
64: 4211, 4302
67: 4946

ALLEN, DAVID, AND OTHERS
65: 1916

ALLEN, DWIGHT W.
67: 3229
68: 7532

ALLEN, MARY P.
68: 7180

ALLEN, RAY
65: 9310
68: 9311

ALLIED HEALTH PROFESSIONS PROJECTS, LOS ANGELES, CALIF.
68: 7625, 7841

ALLINGTON CORP., ALEXANDRIA, VA.
68: 6188

ALSCHULER, ALFRED, AND OTHERS
69: 8366, 8373

ALTMAN, CHARLES DWAYNE
68: 7809

ALTOONA AREA HIGH SCHOOL, PA.
69: 8794

AMARILLO COLL., TEX.
68: 7422

AMBERSON, MAX L., AND OTHERS
69: 10456

AMELON, DONALD JAMES
69: 9184

AMERICAN ASSOCIATION FOR AGRICULTURAL ENGINEERING AND VOCATIONAL AGRICULTURE, ATHENS, GA.
69: 9981, 9982
10426

AMERICAN ASSOCIATION OF DENTAL EXAMINERS
67: 7091

AMERICAN ASSOCIATION OF DENTAL EXAMINERS, ALBANY, N.Y.
69: 6421

AMERICAN ASSOCIATION OF DENTAL SCHOOLS, WASHINGTON, D.C.
68: 7274

AMERICAN ASSOCIATION OF JUNIOR COLLEGES, WASHINGTON, D.C.
nd: 7136
66: 8063
68: 5467
7694

AMERICAN ASSOCIATION OF MEDICAL RECORD LIBRARIANS, CHICAGO, ILL.
65: 5997

AMERICAN ASSOCIATION OF TEACHER EDUCATORS IN AGRICULTURE, ST. PAUL, MINN.
69: 10456

AMERICAN COLLEGE OF DENTISTS
68: 7104

AMERICAN COUNCIL OF INDUSTRIAL ARTS SUPERVISORS
69: 7808

AMERICAN COUNCIL ON EDUCATION, WASHINGTON, D.C.
65: 2236

AMERICAN COUNCIL ON INDUSTRIAL ARTS TEACHER EDUCATION, WASHINGTON, D.C.
67: 2091

AMERICAN DENTAL ASSOCIATION, CHICAGO, ILL.
64: 1532, 1900
66: 2143

PERSONAL AND INSTITUTIONAL AUTHORS INDEX

AMERICAN DENTAL ASSOCIATION, CHICAGO, ILL. (CONTINUED)
67: 5316
68: 9887

AMERICAN DENTAL ASSOCIATION, CHICAGO, ILL. COUNCIL ON DENTAL EDUCATION
65: 4652
66: 1964

AMERICAN DENTAL ASSOCIATION, CHICAGO, ILL. DIV. OF EDUCATIONAL MEASUREMENTS
69: 10058

AMERICAN ENTERPRISE INST. FOR PUBLIC POLICY RESEARCH, WASHINGTON, D.C.
63: 4359

AMERICAN FOUNDATION FOR THE BLIND, NEW YORK, N.Y.
67: 4249

AMERICAN FOUNDATION ON AUTOMATION AND EMPLOYMENT, INC., NEW YORK, N.Y.
69: 10065

AMERICAN HOME ECONOMICS ASSOCIATION, WASHINGTON, D.C.
67: 4817
8083
68: 7114, 7262
69: 9352

AMERICAN HOSPITAL ASSOCIATION, CHICAGO, ILL.
67: 5162

AMERICAN INDUSTRIAL ARTS ASSOCIATION, WASHINGTON, D.C.
66: 5740
67: 1954
5837
69: 7808
9853

AMERICAN INDUSTRIAL ARTS ASSOCIATION, WASHINGTON, D.C. INTERNATIONAL RELATIONS COMMITTEE
69: 9277

AMERICAN INST. FOR RESEARCH IN THE BEHAVIORAL SCIENCES, PALO ALTO, CALIF.
68: 8814

AMERICAN INSTITUTES FOR RESEARCH, PITTSBURGH, PA.
nd: 8526, 8527
65: 1394
66: 1395, 1396, 1397
67: 4848
68: 6859
69: 9451, 8516, 8588

AMERICAN IRON AND STEEL INST., NEW YORK, N.Y.
66: 4684

AMERICAN MEDICAL ASSOCIATION, CHICAGO, ILL.
68: 7497

AMERICAN MEDICAL ASSOCIATION, CHICAGO, ILL. COUNCIL ON MEDICAL EDUCATION
65: 5997
67: 5999
6000

AMERICAN NATIONAL RED CROSS, NEW YORK, N.Y.
66: 8294

AMERICAN NATIONAL RED CROSS, NEW YORK, N.Y. NURSING SERVICES
66: 8295

AMERICAN NURSERYMAN, CHICAGO, ILL.
65: 401

AMERICAN NURSES' ASSOCIATION, NEW YORK, N.Y.
65: 4914, 4915
66: 7326
67: 7327
68: 9033

AMERICAN NURSING HOME ASSOCIATION, WASHINGTON, D.C.
nd: 8321

AMERICAN OCCUPATIONAL THERAPY ASSOCIATION, NEW YORK, N.Y.
67: 3216
68: 8132

AMERICAN PERSONNEL AND GUIDANCE ASSOCIATION, WASHINGTON, D.C.
66: 5436
67: 7493

AMERICAN PHILOSOPHICAL SOCIETY, PHILADELPHIA, PA.
67: 5475

AMERICAN PHYSICAL THERAPY ASSOCIATION, NEW YORK, N.Y.
67: 6688
68: 6693

AMERICAN REHABILITATION COUNSELING ASSOCIATION
66: 5436

AMERICAN SOCIETY FOR ENGINEERING EDUCATION, WASHINGTON, D.C.
62: 4380

AMERICAN SOCIETY FOR PERSONNEL ADMINISTRATION
67: 7343

AMERICAN SOCIETY FOR TRAINING AND DEVELOPMENT, MADISON, WIS.
68: 6940

AMERICAN SOCIETY OF ANESTHESIOLOGISTS, PARK RIDGE,
68: 7497

AMERICAN SOCIETY OF ANESTHESIOLOGISTS, PARK RIDGE, ILL.
68: 7497

AMERICAN SOCIETY OF ANESTHESIOLOGISTS, PARK RIDGE, ILL.AMERICA
68: 7497

AMERICAN SOCIETY OF CLINICAL PATHOLOGISTS, CHICAGO, ILL.
67: 5999
6000

AMERICAN SOCIETY OF PLANNING OFFICIALS
68: 9944

AMERICAN SOCIETY OF RADIOLOGIC TECHNOLOGISTS, FOND DU LAC, WIS.
nd: 7337
67: 8056

AMERICAN SOCIETY OF RADIOLOGIC TECHNOLOGISTS, FOND DU LAC, WIS. EDUCATION COMMITTEE
64: 5840

AMERICAN SOCIETY OF TOOL AND MANUFACTURING ENGINEERS, DEARBORN, MICH.
68: 8762

AMERICAN TECHNICAL SOCIETY, CHICAGO, ILL.
69: 9170, 9494

AMERICAN TELEPHONE AND TELEGRAPH CO., NEW YORK, N.Y.
65: 4682

AMERICAN VOCATIONAL ASSOCIATION INC., WASHINGTON, D.C.
69: 8394

AMERICAN VOCATIONAL ASSOCIATION, INC., WASHINGTON, D.C.
68: 8027
69: 10028

AMERICAN VOCATIONAL ASSOCIATION, WASHINGTON, D.C.
66: 1827
67: 7493
68: 928
7180, 7380, 7814
8026, 8037
69: 9155, 9654, 9748, 9979
10450,10452
70: 10317

AMERICAN VOCATIONAL ASSOCIATION, WASHINGTON, D.C. AGRICULTURAL DIV.
66: 7699

AMERICAN VOCATIONAL ASSOCIATION, WASHINGTON, D.C. AGRICULTURAL EDUCATION DIV.
69: 10454,10455

AMERICAN VOCATIONAL ASSOCIATION, WASHINGTON, D.C. BUSINESS AND OFFICE EDUCATION DIV.
69: 10440

AMERICAN VOCATIONAL ASSOCIATION, WASHINGTON, D.C. COMMITTEE ON PUBLICATIONS
64: 2496

AMERICAN VOCATIONAL ASSOCIATION, WASHINGTON, D.C. COMMITTEE ON RESEARCH IN AGRICULTURAL EDUCATION
63: 2213
68: 8602

AMERICAN VOCATIONAL ASSOCIATION, WASHINGTON, D.C. CURRICULUM MATERIALS COMMITTEE
70: 10509

AMERICAN VOCATIONAL ASSOCIATION, WASHINGTON, D.C. DISTRIBUTIVE EDUCATION DIV.
69: 10432

AMERICAN VOCATIONAL ASSOCIATION, WASHINGTON, D.C. HEALTH OCCUPATIONS SECTION
69: 10434

AMERICAN VOCATIONAL ASSOCIATION, WASHINGTON, D.C. HOME ECONOMICS EDUCATION DIV.
69: 10453

AMERICAN VOCATIONAL ASSOCIATION, WASHINGTON, D.C. INDUSTRIAL ARTS EDUCATION DIV.
69: 10392,10393,10394,10415,

ANNUAL SUPPORTING SERVICE INDEX

VOCATIONAL AND SUPPORTING SERVICES INDEX

AGRICULTURAL EDUCATION
nd: 276, 278
5854
7403, 7405, 7406, 7415,
7417, 7433, 7601, 7843,
7848
8045, 8250
9173, 9282, 9293, 9294,
9298, 9301, 9302, 9309,
9312, 9315, 9318, 9319,
9320, 9321, 9322, 9323,
9324, 9325, 9327, 9328,
9329, 9335, 9338, 9350,
9834
10424
60: 4618
63: 575, 687
1400, 1576
2123, 2213
5271
6502
64: 358, 361, 515, 517,
518, 521, 535, 610,
820, 945
1015, 1443
2309, 2543
3577
4033, 4536, 4538, 4539,
4544, 4545, 4546, 4722,
4723, 4724
6163, 6347
7911
9071, 9078, 9079, 9326
65: 57, 58, 401, 526,
529, 570
1009, 1151, 1373, 1444
2111, 2369
3538
4555, 4562, 4575, 4727
6399, 6426, 6428
8061
9035, 9051, 9252, 9272,
9308, 9310, 9313, 9337,
9341, 9342
66: 372, 373, 374, 375,
376, 377, 383
1131, 1665, 1819, 1949,
1953
3862
4576, 4583, 4584, 4585,
4586, 4840
5186, 5758
6356, 6798, 6823
7165, 7309, 7699
8799
9296, 9300, 9303, 9307,
9314, 9775
67: 475, 971
1457, 1959
2215, 2605, 2609, 2693,
2937
3779
4351, 4644, 4782, 4783,
4784, 4785, 4786, 4787,
4906, 4931
5185
6281, 6282
7160, 7328, 7329, 7416
8503, 8627, 8749
9333, 9336, 9669, 9829,
9843
68: 928
5152, 5545, 5855
6014, 6278, 6336, 6337,
6571, 6725, 6913
7078, 7096, 7181, 7199,
7202, 7284, 7336, 7338,
7351, 7479, 7542, 7557,
7559, 7596, 7597, 7613,
7618, 7619, 7620, 7621,
7628, 7647, 7696, 7698,
7724, 7756, 7809, 7902,
7903, 7930, 7980, 7981
8042, 8074, 8079, 8220,
8252, 8272, 8336, 8602,
8643, 8646, 8656, 8663,
8764, 8766, 8767, 8771,

AGRICULTURAL EDUCATION (CONTINUED)
68: 8822, 8933, 8934, 8935,
8936, 8958, 8984
9077, 9080, 9177, 9287,
9297, 9299, 9311, 9317,
9334, 9339, 9340, 9351,
9772, 9860, 9861, 9999
10000, 10002, 10003, 10004,
10005, 10006, 10007, 10008,
10044, 10045, 10046, 10047,
10048
69: 7965, 7966, 7993
8076, 8299, 8318, 8642,
8765, 8783, 8827, 8828,
8949
9195, 9390, 9392, 9565,
9566, 9567, 9568, 9569,
9610, 9619, 9620, 9621,
9625, 9629, 9630, 9634,
9660, 9668, 9705, 9767,
9778, 9806, 9844, 9858,
9859, 9863, 9871, 9872,
9889, 9898, 9949, 9950,
9956, 9957, 9981, 9982
10001, 10009, 10028, 10035,
10049, 10098, 10116, 10230,
10295, 10302, 10322, 10328,
10370, 10454, 10455, 10456

BUSINESS AND OFFICE EDUCATION
nd: 6189, 6448
7176
8134, 8910
63: 831
3691
6222, 6670
8482
9174
64: 926
1971
5440, 5590, 5821
6745
7205
8658, 8706, 8992
9047
65: 133, 671, 882
1348, 1512, 1599
4028, 4267
9091, 9229, 9251, 9258,
9271
66: 444
2064, 2367
6190, 6396, 6397
7193
8130
9250, 9967, 9968
67: 1577
2234, 2706, 2739
3598, 3640
5917
7032, 7044, 7067
8249, 8840
9191
68: 5935
6720, 6731, 6971
7184, 7186, 7235, 7286,
7432, 7435, 7500, 7561,
7605, 7624, 7653, 7692,
7733, 7777, 7786, 7975,
7976, 7977, 7978
8024, 8057, 8374, 8823,
8839
9053, 9363, 9364, 9365,
9498
69: 8540, 8747, 8913, 8923,
8937, 8939, 8940, 8941,
8942, 8943
9488, 9670, 9673, 9977
10440

DISTRIBUTIVE EDUCATION
nd: 7519
8060, 8069, 8095, 8821,
8824, 8944
9295, 9557
50: 124
52: 123, 152, 173

DISTRIBUTIVE EDUCATION (CONTINUED)
56: 125
57: 126
62: 146, 155, 156, 170
63: 150
2375
8484, 8583
64: 163, 165, 166, 167
1390
2265
3608
65: 209, 210, 215, 423
2898
4682
7690
66: 1621
2493
7518
9270
67: 2617
5973, 5974, 5975, 5976
6011, 6746, 6810
7217
8296, 8599, 8772
10056
68: 7014, 7137, 7246, 7420,
7616, 7697, 7766, 7851,
7952
8021, 8094, 8191, 8195,
8245, 8478, 8479, 8773,
8885, 8938, 8989
9361, 9599
69: 7953
8453, 8454, 8991
9503, 9805, 9877, 9880,
9897, 9951
10401, 10432

HEALTH OCCUPATIONS EDUCATION
nd: 6028, 6145, 6863
7136, 7191, 7337, 7969
8283, 8321, 8418, 8931,
8948
63: 1254
2028, 2114, 2446
4350, 4357
5968
8825
64: 1086, 1532, 1900
4044, 4302, 4339
5149, 5560, 5574, 5840
8616
9090
65: 365, 590
2541, 2555, 2557, 2558,
2599
3221, 3684, 3972
4643, 4652, 4914, 4915
5214, 5358, 5646, 5997
6027, 6127
8763
9036, 9678
66: 1891, 1964
2143, 2393, 2638, 2920
3619, 3641, 3812
4251, 4358, 4556, 4623,
4801
5153, 5600, 5641, 5822
6208, 6254, 6348, 6349,
6413, 6658, 6772
7109, 7326, 7556, 7855
8063, 8162, 8293, 8376,
8461, 8654
9267
67: 192, 771, 818
1078
2138, 2216, 2691, 2888
3216, 3218, 3501, 3615
4085, 4173, 4179, 4240,
4247, 4356, 4624, 4706,
4789, 4790, 4825
5162, 5244, 5316, 5425,
5426, 5427, 5476, 5635,
5954, 5998, 5999
6000, 6314, 6657, 6688,
6697, 6698
7046, 7062, 7091, 7327,

VOCATIONAL AND SUPPORTING SERVICES INDEX

MANPOWER ECONOMICS (CONTINUED)
69: 8780
9245, 9256, 9519, 9759,
9798, 9936
10065, 10099, 10180, 10247,
10414, 10450

OCCUPATIONAL PSYCHOLOGY
64: 5825
65: 2855
66: 2700
3583
7106
67: 4094
7045, 7110
69: 8507

OCCUPATIONAL SOCIOLOGY
nd: 7778
64: 2145
4720
65: 3620
8240
66: 3745, 3756, 3757, 3845,
3860
5225, 5226
6424
7854
67: 3754, 3755, 3758, 3760,
3863
5475
6564, 6624, 6968
7033, 7061, 7066, 7098
68: 5631
6825, 6846, 6862
7048, 7081, 7103, 7146,
7215
8105

TECHNICAL EDUCATION
nd: 5857
6145
10029, 10033, 10034
62: 4380, 4420
63: 4015
64: 515, 592
1559
4518
5270
65: 591
1476, 1524, 1973, 1974,
1988
2301
3857
10443
66: 1942, 1975
2692
3641
4319
8414, 8999
9331
67: 1409
2554
4233, 4792
5610, 5977, 5978
8096, 8334, 8972, 8988
9344
68: 4791
5467, 5628
6858, 6874, 6912
7247, 7414, 7419, 7497,
7561, 7627, 7694, 7756,
7871
8001, 8151, 8176, 8272,
8402, 8486, 8656, 8669,
8870, 8983
9343, 9362, 9379, 9413
69: 7875, 7876, 7961, 7989,
7993
8165, 8318, 8431, 8470,
8472, 8556, 8722, 8786
9163, 9254, 9260, 9370,
9377, 9490, 9526, 9582,
9588, 9603, 9767
10035, 10139, 10140, 10141,
10150, 10151, 10152, 10307,
10435, 10444

TRADE AND INDUSTRIAL EDUCATION
2848

TRADE AND INDUSTRIAL EDUCATION (CONTINUED)
nd: 5579, 5857
7984
8192, 8194, 8291, 8332,
8527, 8553, 8570, 8610,
8613, 8617, 8626, 8805,
8808, 8809, 8810, 8811,
8812, 8911
9284, 9289, 9345, 9615,
9616, 9617, 9756, 9801,
9926
10029, 10032, 10035, 10034,
10037
58: 7394
59: 9592
60: 7396
62: 717
4192, 4380, 4420
6480
7395
63: 2338, 2383
4015, 4800
8806
10031, 10038
64: 432, 592
1339, 1678, 1835
2125, 2297
3189, 3808
4211, 4518, 4611, 4637
5270, 5617, 5717
8154, 8915
9925
10055, 10088, 10089, 10091
65: 591, 861
1619, 1916, 1974
2100, 2397, 2398, 2400,
2401, 2874
3901
4408, 4508, 4600, 4798,
4947
7397
8287, 8288, 8289, 8625,
8959, 8960, 8961, 8985,
8986
9001, 9002, 9003, 9023,
9024, 9025, 9026, 9027,
9028, 9029, 9030, 9034,
9052, 9230, 9305, 9415,
9711, 9737, 9739
66: 1288, 1367, 1552, 1643,
1718, 1942, 1950, 1961,
1975
2593, 2692, 2763
3280, 3554
4799
6484, 6568
7398, 7399, 7400, 7496
8136, 8290, 8414, 8417,
8611, 8624, 8651, 8668,
8857
9331, 9718, 9719, 9720,
9721, 9722, 9723, 9724,
9725, 9726, 9727, 9728,
9729, 9730, 9731, 9732,
9733, 9734, 9735, 9736,
9738, 9740, 9741, 9742,
9743, 9744, 9745, 9746,
9747, 9966
10019
67: 1284, 1409
2214, 2554, 2676, 2678,
2820
3533, 3595
4264, 4392, 4393, 4712,
4773, 4803, 4946
5470, 5597, 5610, 5623,
5640, 5978
6123, 6804
7084, 7101, 7390, 7391,
7392, 7484, 7485
8096, 8334, 8388, 8535,
8807, 8841, 8988
9344, 9374, 9672, 9692,
9707, 9708, 9854
68: 4772, 4791, 4907
5438, 5467, 5639
6194, 6371, 6417, 6519,
6581, 6676, 6678, 6681,
6682, 6739, 6740, 6818,

TRADE AND INDUSTRIAL EDUCATION (CONTINUED)
68: 6838, 6858, 6859, 6874,
6916, 6917
7171, 7185, 7258, 7322,
7349, 7361, 7393, 7461,
7465, 7487, 7504, 7507,
7540, 7584, 7590, 7607,
7629, 7694, 7723, 7752,
7768, 7870, 7871, 7874,
7913
8026, 8027, 8176, 8196,
8312, 8337, 8338, 8339,
8340, 8341, 8342, 8343,
8344, 8345, 8346, 8347,
8348, 8349, 8350, 8351,
8352, 8353, 8354, 8355,
8356, 8357, 8358, 8359,
8360, 8361, 8362, 8402,
8447, 8505, 8569, 8577,
8584, 8623, 8633, 8676,
8714, 8724, 8731, 8838,
8870, 8907, 8983
9022, 9180, 9330, 9343,
9362, 9366, 9367, 9368,
9369, 9371, 9372, 9373,
9375, 9376, 9378, 9379,
9380, 9413, 9671, 9675,
9693, 9709, 9710
69: 7875, 7876, 7961, 7989,
7991, 7998
8271, 8319, 8470, 8472,
8494, 8516, 8536, 8648,
8652, 8666, 8683, 8722,
8925
9176, 9254, 9260, 9277,
9370, 9377, 9582, 9588,
9590, 9814, 9839, 9869,
9890, 9900
10139, 10140, 10141, 10150,
10151, 10152, 10304, 10305,
10431

GENERAL VOCATIONAL AND TECHNICAL EDUCATION
nd: 1463
2641
4352
5483
6143, 6781, 6816
7273, 7295, 7362, 7410,
7661, 7701, 7918
8244, 8300, 8432, 8526,
8573, 8701, 8739, 8849
9395
56: 5637
59: 5256
60: 2135
62: 2839
63: 95, 190
1489, 1669, 1702, 1967
2870, 2947
4035, 4121, 4837, 4950
5178, 5295, 5604
6402
8009, 8442
64: 569
1368, 1411, 1442, 1474,
1553, 1555, 1559, 1624,
1687, 1688, 1722
2496, 2909
3219, 3287, 3288, 3289,
3290, 3524, 3584, 3613,
3826
4275, 4535, 4675, 4679
5603, 5615
6168
8680
65: 477
1029, 1161, 1394, 1436,
1476, 1524, 1606, 1679,
1695, 1728, 1797, 1973,
1988
2006, 2010, 2103, 2104,
2109, 2132, 2191, 2192,
2193, 2194, 2236, 2301,
2359, 2511, 2564, 2574,
2651, 2823, 2854
3685, 3857
4277, 4371, 4424, 4728,

ANNUAL DOCUMENT NUMBER INDEX

VT NUMBER, ISSUE, AND PAGE NUMBER INDEX

VT NO	ISSUE	PAGE
49	ASU70	484
55	SP70	488
57	AP 69	2
58	AP 69	2
89	ASU70	496
94	ASU70	496
95	ASP70	382
98	ASU70	496
107	ASU70	496
117	ASP70	348
118	ASU70	497
120	ASU70	497
121	ASU70	497
123	AP 69	22
124	AP 69	22
125	ASP70	340
126	ASP70	340
133	ASP70	336
135	ASU70	484
146	AP 69	22
150	AP 69	22
152	AP 69	22
155	AP 69	23
156	AP 69	23
163	AP 69	23
165	ASP70	340
166	AP 69	23
167	AP 69	24
170	AP 69	24
173	AP 69	24
176	ASU70	497
190	P 69	38
192	W 69	242
209	AP 69	25
210	AP 69	25
215	AP 69	25
275	ASU70	464
276	ASP70	316
277	ASU70	464
278	ASP70	316
279	ASU70	464
280	ASU70	465
281	ASU70	465
333	ASP70	356
338	ASP70	348
339	ASU70	498
349	ASP70	348
352	ASU70	498
358	ASP70	316
361	ASP70	316
365	W 69	284
367	ASU70	465
368	ASU70	466
372	ASP70	317
373	ASP70	317
374	ASP70	317
375	ASP70	317
376	ASP70	317
377	ASP70	318
383	ASP70	318
385	ASP70	348
386	ASU70	498
387	ASU70	499
388	ASP70	348
389	ASU70	499
392	ASP70	349
398	ASP70	356
401	ASP70	318
404	AP 69	38
423	ASP70	340
432	AP 69	54
444	ASP70	336
445	ASU70	499
446	ASU70	500
447	ASU70	500
458	P 69	2
467	ASP70	356
475	AP 69	2
477	SU70	784
506	ASP70	356
515	ASP70	318
517	ASP70	318
518	ASP70	318
521	ASP70	319
526	ASP70	319
528	ASU70	466
529	W 69	242
535	W 69	232
544	ASU70	466
569	ASP70	382
570	ASP70	319
575	AP 69	2
587	W 69	192
590	ASP70	344
591	ASP70	362
592	ASP70	362
610	ASP70	319
614	P 69	94
671	ASP70	336
679	SP70	442
687	ASP70	320
717	AW 69	212
752	ASU70	524
771	SP70	442
774	P 69	38
816	SP70	560
818	SP70	488
820	AP 69	2
831	ASP70	336
842	AP 69	38
861	AP 69	54
875	ASU70	500
877	W 69	192
882	ASP70	336
888	ASU70	501
889	ASU70	501
890	ASP70	349
891	ASU70	501
892	ASU70	502
893	ASU70	502
898	AP 69	48
926	ASP70	337
928	P 69	94
945	ASP70	320
967	SP70	442
970	SP70	488
971	SP70	570
979	ASP70	356
980	ASP70	357
1009	AP 69	3
1015	ASP70	320
1029	SP70	478
1032	ASP70	349
1036	SP70	478
1078	SP70	442
1086	P 69	38
1087	ASU70	480
1088	ASP70	349
1126	W 69	242
1131	ASP70	320
1151	ASP70	320
1161	AP 69	72
1171	W 69	192
1200	ASP70	357
1211	W 69	192
1231	SU70	792
1254	SP70	478
1258	ASP70	357
1284	AP 69	54
1288	ASP70	362
1289	W 69	300
1315	ASU70	480
1319	SP70	443
1325	ASP70	350
1339	W 69	232
1343	SU70	766
1345	SP70	489
1348	AP 69	14
1367	SP70	489
1368	SP70	560
1373	W 69	242
1390	W 69	193
1394	W 69	232
1395	W 69	232
1396	W 69	233
1397	W 69	233
1400	ASP70	321
1409	W 69	318
1411	SP70	443
1421	W 69	193
1432	ASU70	467
1436	ASP70	382
1441	SP70	489
1442	P 69	76
1443	W 69	338
1444	W 69	338
1449	W 69	193
1455	ASU70	548
1457	W 69	338
1463	SP70	443
1474	W 69	193
1476	SP70	532
1481	SP70	489
1489	W 69	194
1509	SP70	443
1512	W 69	243
1514	SP70	560
1521	P 69	2
1524	W 69	332
1532	P 69	2
1535	AP 69	38
1552	P 69	2
1553	P 69	38
1555	W 69	284
1559	P 69	2
1561	SP70	444
1572	P 69	39
1576	SP70	560
1577	AW 69	183
1599	P 69	3
1606	W 69	194
1614	SP70	548
1619	W 69	194
1621	SP70	444
1624	SU70	746
1643	W 69	318
1646	ASU70	502
1665	AP 69	3
1669	SU70	746
1673	W 69	194

VT NUMBER, ISSUE, AND PAGE NUMBER INDEX

VT NO	ISSUE	PAGE	VT NO	ISSUE	PAGE	VT NO	ISSUE	PAGE
2880	P 69	88	3746	P 69	46	4121	P 69	20
2886	P 69	15	3748	P 69	46	4124	P 69	52
2888	SU70	748	3751	W 69	250	4125	P 69	53
2898	P 69	104	3752	P 69	46	4150	ASU70	524
2909	P 69	16	3753	P 69	47	4173	W 69	199
2916	P 69	16	3754	P 69	47	4179	P 69	53
2920	P 69	16	3755	P 69	77	4187	W 69	296
2937	AW 69	162	3756	P 69	47	4192	AP 69	55
2947	P 69	16	3757	P 69	77	4196	W 69	199
3189	AP 69	55	3758	P 69	77	4211	P 69	104
3216	W 69	196	3760	W 69	250	4233	W 69	199
3218	W 69	246	3773	P 69	19	4240	W 69	256
3219	P 69	41	3779	W 69	326	4247	P 69	20
3221	P 69	41	3782	P 69	47	4249	P 69	53
3229	P 69	17	3796	AP 69	72	4251	P 69	20
3234	W 69	246	3800	W 69	250	4264	AP 69	55
3245	P 69	17	3801	W 69	251	4267	W 69	199
3280	P 69	17	3802	W 69	251	4275	P 69	20
3287	P 69	42	3808	W 69	251	4277	W 69	322
3288	P 69	42	3812	W 69	252	4301	P 69	84
3289	P 69	42	3814	W 69	252	4302	W 69	339
3290	P 69	42	3817	W 69	252	4319	P 69	20
3291	W 69	246	3826	W 69	252	4322	P 69	53
3293	W 69	247	3833	P 69	48	4323	P 69	54
3312	P 69	76	3834	P 69	48	4339	W 69	339
3314	P 69	42	3835	W 69	253	4341	W 69	200
3362	W 69	247	3836	P 69	48	4342	P 69	21
3363	ASP70	382	3837	P 69	48	4350	P 69	21
3372	W 69	247	3838	P 69	49	4351	W 69	257
3409	P 69	43	3839	P 69	49	4352	SU70	832
3433	SU70	766	3840	P 69	49	4353	SU70	784
3435	W 69	248	3841	P 69	49	4356	W 69	257
3496	P 69	43	3842	W 69	253	4357	P 69	21
3501	SP70	447	3843	P 69	50	4358	P 69	22
3505	P 69	76	3844	W 69	253	4359	P 69	54
3521	AP 69	38	3845	W 69	254	4360	P 69	22
3523	P 69	18	3846	W 69	254	4371	P 69	54
3524	W 69	285	3849	P 69	50	4372	W 69	301
3533	AW 69	213	3850	W 69	254	4375	P 69	54
3538	P 69	43	3852	W 69	254	4380	ASP70	363
3554	P 69	18	3857	P 69	50	4392	P 69	84
3570	SU70	766	3860	W 69	255	4393	SU70	748
3577	W 69	248	3862	P 69	51	4401	SU70	748
3583	W 69	248	3863	W 69	255	4408	W 69	257
3584	P 69	18	3868	P 69	77	4419	W 69	235
3587	ASU70	503	3873	W 69	300	4420	P 69	55
3595	SU70	767	3875	P 69	51	4424	W 69	257
3598	AW 69	182	3876	P 69	51	4503	P 69	55
3608	AP 69	26	3882	W 69	255	4508	W 69	200
3613	P 69	18	3883	W 69	256	4509	P 69	55
3615	SP70	552	3893	P 69	51	4510	P 69	55
3619	P 69	19	3900	SU70	767	4518	W 69	200
3620	W 69	249	3901	SU70	793	4519	W 69	201
3640	W 69	197	3902	SU70	793	4521	P 69	56
3641	W 69	197	3903	W 69	256	4526	W 69	201
3648	P 69	44	3942	SP70	570	4532	W 69	201
3649	P 69	44	3960	P 69	52	4535	P 69	112
3650	P 69	44	3961	P 69	52	4536	P 69	112
3652	P 69	45	3962	P 69	52	4538	W 69	201
3668	P 69	45	3972	SP70	495	4539	W 69	258
3669	W 69	249	3974	SU70	767	4544	W 69	258
3684	P 69	45	4015	SU70	762	4545	W 69	202
3685	W 69	197	4019	W 69	332	4546	W 69	285
3691	W 69	198	4026	W 69	198	4555	W 69	258
3693	W 69	249	4028	AW 69	182	4556	P 69	78
3697	P 69	45	4033	AW 69	162	4558	ASU70	467
3700	P 69	19	4035	W 69	198	4562	W 69	318
3719	W 69	249	4044	W 69	198	4563	W 69	259
3723	SU70	793	4085	SP70	532	4570	P 69	56
3745	W 69	250	4094	W 69	256	4572	P 69	58